TENOR 2017

INTERNATIONAL CONFERENCE ON TECHNOLOGIES FOR MUSIC NOTATION & REPRESENTATION

UNIVERSITY OF A CORUÑA

SPAIN

Proceedings of the
Third International Conference on
Technologies for Music Notation and Representation
TENOR 2017

Helena Lopez Palma,
Mike Solomon,
Emiliana Tucci,
Carmen Lage
(editors)

A Coruña 2017

Universidade da Coruña
Servizo de Publicacións

Proceedings of the Third International Conference on Technologies for Music Notation and Representation TENOR 2017
http://www.udc.es/grupos/ln/tenor2017/index.html

LOPEZ PALMA, Helena; SOLOMON, Mike; TUCCI, Emiliana; LAGE, Carmen
A Coruña, 2017
Universidade da Coruña, Servizo de Publicacións
University of A Coruña, UDC Press.

Number of pages: viii+239
210x197 mm
index: p. vii

ISBN: 978-84-9749-666-7
Persistent URL: http://hdl.handle.net/2183/18494

CDU: 781.24:004.4(100)(042.3)*TENOR2017
IBIC: AVQ | UYU | GTN

Edition
Universidade da Coruña,
Servizo de Publicacións: www.udc.gal/publicacions

Design and Layout: NATURAL LANGUAGE UNIT
LaTeX editors: Natural Language Unit
using LaTeX's 'confproc' package, version 0.8 (by V. Verfaille)

Logo design: Nicolas Taffin

UNIVERSIDADE DA CORUÑA

Natural
Language
Unit

Deputación
DA CORUÑA

TENOR 2017 – The Third International Conference on Technologies for Music Notation and Representation

The goal of the Third International Conference on Technologies for Music Notation and Representation has been to address a set of specific research issues associated with Music Notation that were elaborated at the first two editions of TENOR: the foundational Paris conference and the second one held at Cambridge. The Coruña TENOR conference has aimed at focusing the scientific and artistic inquiry over music notation on two themes: vocal music and digital archives. During TENOR 2017, 3 workshops, 1 keynote conference, 21 presentations, and 2 concerts have offered an overview of the state of the art in the representation of music creation.

The three pre-conference workshops focused on innovative technological approaches to music notation: "Hacking, extending and wrapping" by Mike Solomon; "The Guido Language and Engine" by Dominique Fober and Mike Solomon; and "Totalitarian Scoring Workshop" by Benedict Eric Carey and Ryan Ross Smith.

The Keynote conference by Goffredo Haus and Luca Ludovico — "Digitization of Historical Music Archives: Preserving the Past, Embrazing the Future" — presented the digitization technology created by the Laboratorio di Informatica Musicale of the Department of Computer Science of the University of Milan to build the LIM Music Archive, which includes among its digital collections music scores, libretti, graphical documents and objects associated to opera performances taken place at La Scala of Milan.

Papers focused on new forms of symbolic representation of music events including electronic music, interactive performances, live coding, as well as the migration of musical instruments to gestural and mobile platforms, hybridisations with dance, 3D design and multimedia. They presented innovative techniques of music notation, transcription and sonic visualisation which are associated to the fields of musical analysis, composition, performance and acoustics. Paper presentations were articulated in 8 sections: Music transcription, graphic notation, visualisation and analysis, ontology, voice, interaction, and music collaboration.

Concert 1 included five world premier compositions authored by Richard Hoadley: "Homage to Cervantes"; Tomas Marco: "Persiles avista Roma"; Helena Palma: "Voice prints"; Seth Shafer: "Terraformation"; and Silvia Teles: "Des pas sur l'invisible"; and a presentation by soprano Kristina Warren. Hoadley's and Marco's compositions were commissioned by TENOR 2017 as part of the commemorative events set to homage Cervantes on his 4th Centenary. Concert 2 included a world premier of the collaborative virtual reality composition created at the workshop "Totalitarian Scoring", and the interactive music game, "Plurality Spring", designed by Paul Turowski.

The workshops took place at the Faculty of Philology in the University of A Coruña. The scholarly conference, papers, music sessions and concerts took place at Paranymph Hall of the Rectorate of the University of A Coruña.

TENOR 2017
Organising Committee

Organizing Committee

Helena Palma (UDC)
Mike Solomon (Ensemble 101)
Amor Admella (UDC)
Laura Castro (UDC)
Felix Cordoba (UDC)
Manuel Fernandez Ferreiro (UDC)
Mercedes Fernandez Valladares (UCM)
Fernando Gonzalez (UDC)
Andres Lacasa Nikiforov (OSG)
Carmen Lage (UDC)
Teresa Lopez (UDC)
Maria Jesus Lorenzo (UDC)
Carmen Parrilla (UDC)
Nieves Pena (UDC)
Mercedes Regueiro Diehl (UDC)
Dolores Sanchez Palomino (UDC)
Xose Manuel Sanchez Rei (UDC)
Emiliana Tucci (UDC)

Steering Committee

Jean Bresson (IRCAM STMS)
Pierre Couprie (IReMus Univ Paris-Sorbonne)
Dominique Fober (GRAME)
Yan Geslin (INA-GRM)
Richard Hoadley (ARU)
Mike Solomon (Ensemble 101)

ScientificCommittee

Trevor Bača
Karim Barkati
Sandeep Bhagwati
Alain Bonardi
Bruno Bossis
Jean Bresson
Pierre Couprie
Cecile Davy-Rigaux
Carlos Duque
Elsa De Luca
Dominique Fober
Jason Freeman
Ichiro Fujinaga
Yann Geslin
Daniele Ghisi
Gérald Guillot
George Hajdu
Goffredo Haus
Richard Hoadley
Cat Hope
Adrian Hull
David Kim-Boyle
Peter McCulloch
Thor Magnusson
Tomas Marco
Chris Nash
Helena Palma
Jean Penny
Philippe Rigaux
Eleanor Selfridge-Field
Mike Solomon
Mathew Thibeault
Lindsay Vickery
Hasnizam A. Wahid

Performers

Alfredo Garcia (baritone)
Adrian Pais (saxophone)
Florian Vlashi (violin)

Conference Program

DIGITIZATION OF HISTORICAL MUSIC ARCHIVES: PRESERVING THE PAST, EMBRACING THE FUTURE

Goffredo Haus, Luca A. Ludovico
Laboratorio di Informatica Musicale
Dipartimento di Informatica "Giovanni Degli Antoni"
Università degli Studi di Milano
Via Comelico, 39 - 20135 Milano - Italy
{goffredo.haus,luca.ludovico}@unimi.it

ABSTRACT

Cultural institutions dealing with music (opera houses, conservatories, public and private collections, etc.) often hold huge archives made of music-related heterogeneous materials. These subjects can greatly benefit from digitization campaigns and the consequent adoption of ICT techniques as it regards not only the preservation, but also the exploitation and revivification of their content. This paper, that summarizes the keynote speech held at the 3rd International Conference on Technologies for Music Notation and Representation (TENOR 2017), starts from the experiences of the Teatro alla Scala and the Ricordi Historical Archive in order to show the new possibilities emerging from the adoption of computer-based technologies and approaches.

1. INTRODUCTION

Opera houses, conservatories, public and private collections and other music institutions often hold in their archives an invaluable heritage made of heterogeneous materials, including scores, audio recordings, iconographic material, books and letters, etc. In these contexts, a digital vision – mainly based on digitization campaigns and the consequent adoption of technological approaches and methodologies – is becoming increasingly important, not only for the preservation, but also for the exploitation and reliving of their music and music-related content.

During more than 30 years of activity, the Laboratorio di Informatica Musicale (LIM) [1] of the University of Milan has been carrying out international projects and establishing collaborations with important music institutions, such as: Bach Archiv Leipzig, Bolshoi Theatre of Moscow, Orchestra Verdi di Milano, RAI Radiotelevisione Italiana, Ricordi Historical Archive, RSI Radiotelevisione Svizzera, and Teatro alla Scala of Milan.

The goal of this work is twofold: on the one side, summarizing the main lessons learned during the past experiences involving both technologies and music; on the other side, showing the new possibilities and the practical implications of advanced computer-based approaches applied to music archives.

[1] In English: Laboratory of Music Informatics.

2. THE HISTORICAL ARCHIVE OF THE TEATRO ALLA SCALA

As a relevant example of cultural institution holding a rich and heterogeneous archive, it is worth citing the case of the Teatro alla Scala of Milan, often briefly referred to as La Scala. Inaugurated on 3 August 1778, during the last two centuries it hosted the greatest opera singers, dancers, soloists and conductors, and it premiered renowned operas such as *Nabucco*, *Otello*, *Falstaff*, *Madama Butterfly* and *Turandot*.

In addition to pursuing artistic activities and cultural dissemination, the theater has also the key mission of preserving and exploiting the immense amount of music-related materials collected during its history. Such a cultural treasure embraces not only scores, but also recordings, photos, sketches, technical drawings, fashion plates, craft-made objects, and an intangible heritage of human skills and competences.

In recent times, thanks to the *ArchivioLaScala* web site, [2] the theater has provided enthusiasts and scholars with the opportunity to easily access its history and artistic heritage. This initiative is the public and tangible result of a 20-years-long effort involving a number of aspects: an extensive digitization campaign for thousands of materials physically preserved in the archives, a change in the operating procedures adopted by the technical staff in order to produce digital content, and the use of ad-hoc computer technologies to interact with the new platform, as explained below.

2.1 La Scala DAM

In order to preserve its cultural heritage and enhance its fruition by artists, technicians and scholars, in 1998 the superintendence of the theater decided to build an integrated asset management system, called *La Scala DAM*. [3]

[2] URL: http://www.archiviolascala.org

[3] DAM was conceived as a multi-lingual acronym standing for *Digital Asset Management* in English, and for *Depositi, Archivi e Magazzini* (i.e. repositories, archives and warehouses) in Italian.

Figure 1. The Heraeus UT6200 oven used to thermally pretreat open-reel magnetic tapes.

Figure 2. Revox B77 and Otari MX-55 tape recorders.

The idea was to create the digital archive of all available content stored in different locations – i.e. to integrate several uncorrelated archives of the theater – in order to cover a timespan from the second decade of the 20th century to the present day.

The digitization campaign began in 1996 with the preservation of the phonic archive, a project carried out in collaboration with the LIM. As a result, about 5000 open-reel magnetic tapes since 1950 were preserved and restored, for a total amount of about 10,000 hours of audio. Analog media were thermally pretreated to recover the original audio information [1] thanks to a laboratory oven with forced convection (see Figure 1), and then digitized through ad hoc equipment (see Figure 2). In 2000, a similar project was carried out by the LIM in order to rescue about 200 magnetic tapes of high historical interest coming from the phonic archive of the Bolshoi Theatre of Moscow [2].

Since 1998, the digitization project was extended with a modular approach to all the departments of the theater, including *Scores*, *Costumes and accessories*, *Sketches and fashion plates*, *Photos and playbills*, *Properties*, and *Editorial* archives [3].

In this context of archive integration, the LIM research team proposed an advanced multimedia object-relational architecture to query and retrieve musical information in a multimodal way. The idea was to provide access to audio recordings and to the corresponding scores within an integrated environment, supporting different tools (e.g., queries by humming, symbolic inputs, etc.) to search in the digital repository. This platform, called *Musical Archive Information System* (*MAIS*), was profitably experimented at La Scala and documented in several scientific works. The *MAIS* introduced some innovative features with respect to the state of the art, such as different abstraction levels for the description of music information, multimedia content integration, and both symbolic and audio queries to retrieve data from the database. For further details, please refer to [4].

La Scala DAM was initially designed as a restricted-access application running on La Scala's local network for theater workers and artists, with a very limited data exchange from and to the official web site. The project was internally released in 2006, making a huge amount of digital content available to technicians, artists and employees for the creation, production and documentation of each show.

Conversely, nowadays most materials are publicly available on line. In this way, the theater not only responds to the mission of cultural dissemination, but also tries to involve and gain a new audience through multimedia content sharing.

The management of the project was entrusted to a group of institutional, technological and scientific partners, including Fondazione Milano per la Scala, University of Milan, Accenture, Fastweb, Hewlett-Packard, Oracle, and TDK.

At the moment of writing, the available digital contents cover approximatively:

- 17,000 posters and playbills documenting all the artistic activities of the theater, including operas, ballets, concerts, recitals, and other cultural events. These materials are scanned at a high resolution, and their content (i.e. the schedule, the playlist, and the cast of the show) is manually transcribed;

- more than 1,000,000 on-stage and back-stage photos, portraying renowned singers (Carlo Bergonzi, Renato Bruson, Maria Callas, Mario Del Monaco, Giuseppe Di Stefano, Placido Domingo, Mirella Freni, Tito Gobbi, Luciano Pavarotti, Giulietta Simionato, Joan Sutherland, Renata Tebaldi, etc.), conductors (Claudio Abbado, Daniel Barenboim, Pierre Boulez, Guido Cantelli, Victor De Sabata, Gianandrea Gavazzeni, Carlo Maria Giulini, Carlos Kleiber, Riccardo Muti, Arturo Toscanini, Herbert von Karajan, etc.), dancers and choreographers (Roberto Bolle, Alessandra Ferri, Carla Fracci, Rudolf Nureyev, Luciana Savignano, etc.);

- 24,000 sketches, drawings and fashion plates signed by great artists, including Alberto Burri, Marc Chagall, Jean Cocteau, Salvatore Fiume, Piero

Fornasetti, Renato Guttuso, Marino Marini, Pablo Picasso, and Mario Sironi;

- 45,000 costumes prepared by designers such as Caramba, Emanuele Luzzati, Vera Marzot, Odette Nicoletti, and Franca Squarciapino;
- 60,000 costume accessories, including jewelery, lingerie, shoes, wigs and hats;
- 80,000 scene tools.

Since the theater is still active and the project running, the mentioned numeric data are growing day by day.

2.2 Technical Challenges and Remarks

The technological core of *La Scala DAM* project is a multimedia relational database, with a number of applications attached to feed the database and to retrieve information. Two goals guided the project: i) information entities had to be described in the most detailed way according to the models already in use in the theater, and ii) a network of relationships as rich as possible had to be established among those entities.

A first result is to provide a comprehensive description of a given object from multiple perspectives. This concept can be declined at different levels of aggregation and abstraction. For instance, if we focus on a single physical good, say a costume, the concept of "comprehensive description" may embrace its data sheet, the list of available accessories, the original fashion plate, a number of on-stage photos, and so on. Conversely, if we take into consideration an opera, this approach implies the possibility to easily obtain a synoptic view of all information somehow related to this work.

When the network of relationships is properly exposed through an ad-hoc application, it can provide the user with multiple navigation paths towards the requested information. An example is the following list of operations: search for a costume, retrieve the opera it was designed for, move to the list of all on-stage representations for that opera, find the name of the artist who played the role of the protagonist in a given date, watch the list of all the operas he/she has performed in the last season, choose one of those operas and open all the related photos, select a photo, and finally open its detailed data sheet.

These considerations guided the design of the web interface to browse content from all archives in an integrated way. Figure 3 shows some screenshots from the original web application. Much of the interface – not only icons and images but also most text parts – could be clicked, allowing to jump to new lists of object or to other meaningful aggregations of information.

During the project, not only technical issues but also relational problems emerged, mainly related to the paradigm shift required to the staff of "traditional" archives, as well as to the perceived feeling of autonomy loss in sectors managed independently so far.

Being the theater in activity, the design, implementation and testing of the solution did not have to affect the operating procedures of the theater staff, nor it had to hinder the

Figure 3. Screenshots from *La Scala DAM* web application. From top to bottom: the synoptic view of music works, the list of playbills, the list of photos.

production environment. The risk was the realization of a very powerful platform either unusable due to its complexity or poorly used because of the extra workload required to archive employees. For this reason, all the phases of the project were conducted in tight cooperation with the theater governance, archive managers, domain experts and workers, in a continuous exchange of critical observations and improvements.

During the digitization phase, a key problem was how to catch and represent physical objects. Needless to say that – in a multimedia database – only digital representations can be entered. Some archive contents were already in digital format, for instance recent photos and recordings as well as computer processed texts; but other materials had to undergo an analog-to-digital conversion, and in this case the main concern was to prevent information loss, sometimes in absence of international guidelines to follow. The real challenge concerned the digitization of 3D physical objects, [4] whose appearance and features could not be digitized with the available technologies. In this case, the digital representation could catch only some aspects considered relevant by experts. Let us cite the example of a stage tool or a costume, where multiple photos can capture shapes, colors and design details from different angles, but they do not allow a 3D reconstruction nor they provide information about materials.

Finally, the main question was how to structure information inside the database. An opera house like La Scala runs different kinds of performances, ranging from operas, ballets, and symphonic concerts to interviews, conferences, and public presentations; and each activity presents peculiar features concerning its structure, staging, and relationship with other information entities. Since the database had to cover all possible cases, the problem of information structuring was not a trivial one.

This issue was solved adopting a 3-tier hierarchy starting from the concept of *base version*, namely a work as it was conceived by its author(s). This definition is meaningful for operas, ballets, and concerts, but – clearly – it makes no sense for activities such as conferences or presentations. A base version has a number of metadata attached, including title, author names and roles, premiere date and place, and ensemble. The second key concept is the one of *staging*, that clusters a set of performances characterized by the same music program, belonging to the same season and having common features about production, staging, and cast. For example, "Le nozze di Figaro" by W.A. Mozart and L. da Ponte is a base version, whereas the 2005/06 production of that opera is a staging. In general terms, the same base version can have 1 to n stagings attached. Finally, the 3-tier hierarchy is closed by the concept of *performance*, namely the instance of a base version according to a particular staging, in a given date and place and with a specific cast. At La Scala, performances present a biunivocal relation with the corresponding playbills. Moving from base version to staging and performance, information becomes more and more detailed.

Such a structure is suitable to represent also non-standard (yet frequent) cases. First, a single performance can include many independent music works, like in singing recitals, typically made of arias from different operas. This situation is managed by linking each subpart of the staging to a different base version, which requires an n to m relationship between the two entities in the database schema. Another issue is how to represent non-musical events, a case that can be easily solved by creating dummy base versions.

After defining the 3-tier spine, other entities are put in relationship with the mentioned concepts in the most proper way. For example, librettos and synopses are naturally linked to base versions, fashion plates and costumes to stagings, and audio recordings and on-stage photos to performances.

A simplified entity-relationship diagram for the resulting database is shown in Figure 4. For a more detailed discussion, please refer to [5].

[4] Also analog scores, photos and playbills can be considered as physical objects, but – in general – their information content can be fully captured by scanning a single 2D side.

3. THE RICORDI HISTORICAL ARCHIVE

Ricordi is an Italian music publisher that has promoted famous composers and musicians over more than two centuries of activity. Today, its Historical Archive is probably the most important private music collection in the world, including invaluable treasures such as the autograph scores of many operas by G. Verdi and G. Puccini. The collection embraces handwritten scores, printed scores, librettos, photographs, drawings, posters, letters, periodicals, and administrative documents.

Compared to the case of La Scala, the "core business" of the Ricordi Historical Archive is different: artistic production for the former, cultural heritage preservation for the latter. For this reason, the Archive produces no audio content (even if a collection of historical vinyl recordings is currently being bought), and its heritage is not going to grow significantly over time.

In the first decade of the new millennium, the Italian ministry for cultural heritage and activities (Ministero per i Beni e le Attività Culturali, MiBAC) promoted and funded an extensive digitization project involving the most important Italian musical institutions. The aim was to create the Italian music network (*Rete della Musica Italiana*, *ReMI*), a new management and diffusion data architecture focusing on Italian musical heritage. This goal was achieved by connecting peripheral repositories of music-related institutions each other and providing web users with centralized query services. On the one side, local repositories maintained their independence concerning the organization and management of their own contents, receiving public funding to adhere to the project; on the other side, centralized web services integrated and homogenized data access, guaranteed service continuity, optimized answering time, and offered unified user-friendly interfaces.

Launched on 20th June 2008, the new music-search interface was published within a thematic area of the Ministry's portal known as *InternetCulturale*, [5] and offered a synoptic view and an integrated navigation of music documents preserved across Italy [6].

In order to support interoperability among heterogeneous systems, the digitization and cataloging of documents had to adhere to international standards, and specifically:

- Universal Machine Readable Catalogue (UNIMARC) [7], Dublin Core and Management Administrative Metadata (MAG) for metadata encoding;
- Open Archival Information System (OAIS) [8] and Open Archive Initiative Protocol for Metadata Harvesting (OAI-PMH) [9] for preservation, interoperability and sharing models.

The LIM was in charge of the scientific coordination of the project, including the study of ad-hoc standards for cataloging music-related materials, the design and implemen-

[5] URL: `http://www.internetculturale.it`

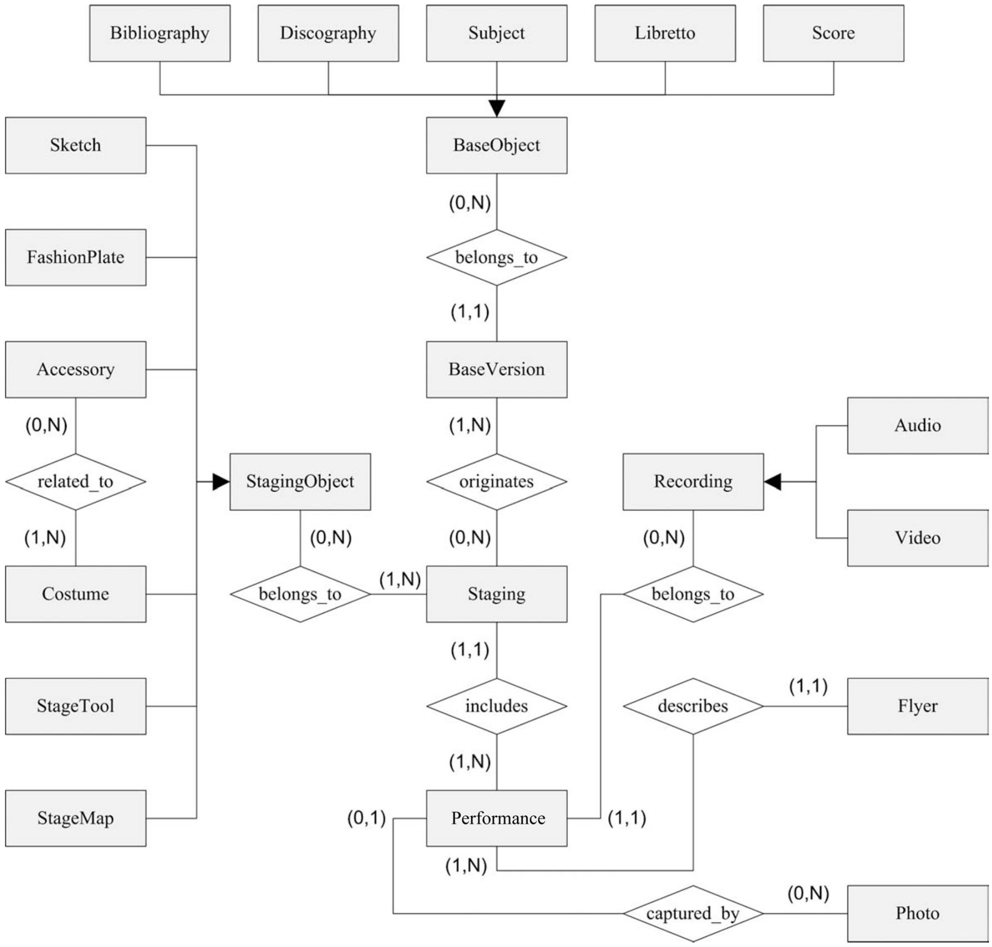


Figure 4. Entity-relationship diagram of *La Scala DAM* database.

tation of the integrated multimedia database, and the realization of the central web application.

In this context, MiBAC and Ricordi signed an agreement to make part of the Archive's content publicly available. The digitization project involved a total amount of 6,586 documents (scores, texts, photos, drawings, sketches, fashion plates, scenic maps, etc.) and resulted in 12,660 digital scans related to about 100 music works. Digital scans were produced with high quality settings and saved in formats adequate for preservation and publishing purposes, and subsequently down-sampled and saved for web browsing. All the materials were cataloged by experts in compliance with international standards.

After the conclusion of the publicly funded project, the digitization and cataloging campaign was carried on at a local level, thus creating the current Ricordi digital archive. [6]

The structure of both the peripheral (Ricordi) and the central (*ReMI*) database, as well as the design of the corresponding web applications to browse multimedia content, were clearly inspired by the experience made at La Scala. Similitudes between the two approaches are evident in the screenshots in Figure 5, extracted from *InternetCulturale*. The main difference concerns the kind of materials locally available, including for example handwritten scores instead of audio content. For the sake of clarity, it is worth underlining that – in the wider framework of *ReMI / InternetCulturale* – the idea was to integrate heterogeneous information and digital objects coming from different archives, so the lack of audio resources from Ricordi had to be filled by other institutions.

While maintaining the same approach, the design phase of the Ricordi platform was influenced by the different mission that a historical archive has with respect to an opera house in activity. In the former case, the goal is to highlight the richness and variety of the preserved heritage, to give new value to archive materials, to disseminate musical culture by providing easy access to sources, also through the interoperability with other platforms; in the latter case, the focus is on gaining new audience and supporting the local artistic production. These different approaches influence many aspects of digitization, indexing and browsing, from the compliance with international standards to the design of meaningful relationships oriented to specific navigation paths.

An interesting evolution of Ricordi's digitization project – mainly conducted on graphic materials (see Figure 6) – is the reliving of its tangible and intangible cultural heritage through international exhibitions. These initiatives propose a mix among historical materials presented to the public in their original form and innovative technological

[6] URL: `http://digital.archivioricordi.com/`

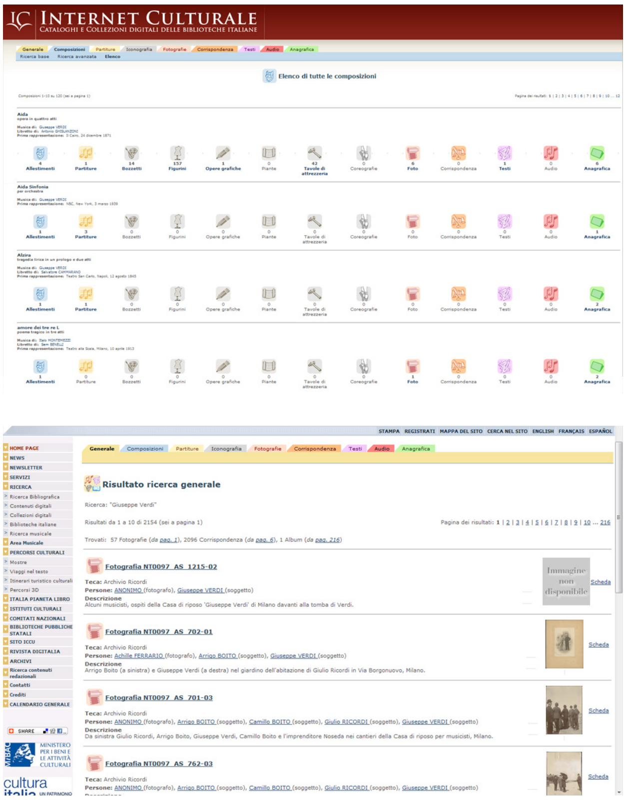


Figure 5. Screenshots from the *InternetCulturale* web portal.

applications to add multimedia interactivity (see Figure 7). The availability of a digital catalog simplifies the organization of cultural events focusing on a specific theme (e.g., an author, a music work, a place, etc.), and a comprehensive library of digital objects makes it particularly easy to produce promotional and editorial materials and to implement computer-based applications.

The list of exhibitions organized by Ricordi in partnership with the LIM includes *Celeste Aida* (November 2006), *That's Opera* (November 2008), *That's Butterfly* (September 2009), *The Enterprise of Opera* (August 2013), and *Madama Butterfly - L'oriente ritrovato* (November 2016).

4. LEARNING FROM THE PAST

If we look back on the experiences conducted at important music institutions and archives, the outcomes are mainly satisfactory. There is a number of activities that have greatly benefited from digitization, computer-based approaches and technological innovations. Just to name a few examples:

- The adoption of correct procedures to ensure the preservation of cultural heritage;
- The application (or even the pioneering research) of standards for cataloging and sharing, with the side effect of fostering the interoperability with other systems;
- The experimentation of new paradigms to query and browse metadata and multimedia objects in an integrated way, thus improving the access to information and – consequently – facilitating organization, production, and communication processes;
- The release of innovative applications based on multimodal representations of music and music-related information.

Unfortunately, the introduction of computer-based approaches into "traditional" environments often implies additional work and requires the investment of extra resources, and – in these cases – long-term advantages need to be seen over short-term disadvantages. We can mention a number of real cases where innovation was not retained as an opportunity but as an obstacle:

- The database management system originally conceived by the LIM for *La Scala DAM* was object-oriented (OODBMS), but it was considered too advanced and somehow experimental by the other technological partners, so the idea was abandoned in favor of a more standard relational model;
- At La Scala, the new procedures oriented to a digital management of information, enthusiastically welcomed by those archives used to work with digital objects (e.g., the *Photos and playbills* and the *Phonic* archives), were initially rejected by some sectors of the theater (e.g., the *Sketches and fashion plates* archive);
- Sometimes, the experimentation of cutting-edge technologies (e.g., the *MAIS*) was not adequately appreciated, as people focused on the additional burden (e.g., the encoding of scores into a symbolic format) without seeing their potential (e.g., the possibility to query a huge archive of music works by humming or playing a digital instrument).

Experience teaches us that one of the most demanding tasks for computer experts is to clearly and convincingly illustrate the undeniable long-term benefits of ICT approaches and techniques.

Another cultural challenge is to explain that digitization does not imply preserving forever. A continuous investment of resources is required to ensure that: i) the media remain intact (the solution is to perform periodical tests and new copies, when needed), ii) the encoding formats continue to be known, documented and readable (open and commonly accepted standards are the best option, re-coding is an alternative), and iii) the devices to read them can be kept running or – in case – replaced.

5. SETTING OUT THE FUTURE

A cultural institution such as La Scala is economically supported by the income from its multiple activities (ticket sales, space and equipment rental, broadcasting rights and contracts, merchandising, etc.), not to mention public and private funding. Conversely, the economic sustainability of an archive is often based on the exploitation of its assets. From this point of view, the business model adopted by the

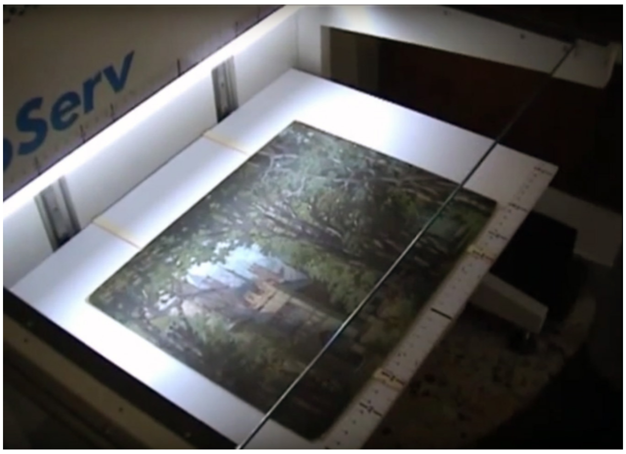


Figure 6. The ProServ ScannTECH 601c used to digitize Ricordi's materials.

Figure 7. *The Enterprise of Opera* exhibition in Berlin. In the foreground, an application to interact with music content based on the IEEE 1599 standard.

Ricordi Historical Archive is paradigmatic: the organization of exhibitions and events, the participation in national and international funded projects, editorial initiatives and sponsorships are the means to support the preservation of cultural heritage and to fulfill the mission of dissemination. This approach requires not only to showcase the available cultural assets, but also to find new ways to relive them. In this sense, information and communication technologies can play a key role.

A first way to benefit from digitization and new technologies is to integrate heterogeneous information according to a multilayer model, thus adding new value to the original uncorrelated objects. For the mentioned archives, aggregations led to a better access to information and to a more comprehensive view on assets. Thanks to ad-hoc file formats and computer applications, it is possible to extend such an approach from databases to other contexts. For example, music-related cultural heritage can be experienced adding synchronization features and interaction with multimedia content. These concepts have been applied in a number of Ricordi's exhibitions, where computer-based interfaces let the audience browse autograph scores, follow music notation on different score editions, and compare historical audio performances through interactive stations, designed also for musically-untrained people. To that end, the cooperation between the Ricordi Historical Archive and the LIM suggested the adoption of the IEEE 1599 technological framework [10]. By integrating different and heterogeneous aspects of a music piece within a unique document, the IEEE 1599 standard creates a sort of local semantic network realized through a multilayer structure [11], which in turn can be integrated into a global network such as the Semantic Web. This approach fosters a more advanced music experience and allows the implementation of innovative services, such as interactive playbills or the augmented fruition of a music show [12].

The integration – at a local and global level – of related content may provide new value to the original uncorrelated entities, potentially introducing new intellectual-property rights to be protected and exploited [13].

An interesting question for cultural heritage holders is whether technologically advanced approaches are pure scientific research and experimentation, or they can lead to the release of marketable products. An answer is provided by the mentioned experiences: both La Scala and Ricordi have improved their ability to go on the market, for example by attracting a new audience through their web archives, offering their goods and services (staging rentals, exhibitions, etc.) in a more effective way, and publishing editorial products based on digitized materials.

Concerning the revivification of archive content, many other successful examples could be mentioned: from interactive and adaptive products for music education [14], like a textbook recently published by Pearson, to new models to interact with live music performances [15].

By combining the efforts of cultural heritage holders and computer scientists, it is possible not only to create new value for music archives but also to perform a relevant cultural operation based on innovative models to experience music.

6. REFERENCES

[1] G. Haus, "Rescuing La Scala's music archives," *Computer*, vol. 31, no. 3, pp. 88–89, 1998.

[2] G. Haus and M. L. P. Pajuelo, "Music processing technologies for rescuing music archives at Teatro alla Scala and Bolshoi theatre," *Journal of New Music Research*, vol. 30, no. 4, pp. 381–388, 2001.

[3] G. Haus, A. Paccagnini, and M. Pelegrin, "Characterization of music archives' contents. a case study: the archive at Teatro alla Scala," in *Proc. of the 3rd International Congress on Science and Technology for the Safeguard of Cultural Heritage in the Mediterranean Basin, Alcalà de Henares, Spain*, 2001.

[4] E. Ferrari and G. Haus, "The musical archive information system at Teatro alla Scala," in *Multimedia Computing and Systems, 1999. IEEE International Conference on*, vol. 1. IEEE, 1999, pp. 817–821.

[5] G. Haus and L. A. Ludovico, “The digital opera house: an architecture for multimedia databases,” *Journal of Cultural Heritage*, vol. 7, no. 2, pp. 92–97, 2006.

[6] L. Ciancio, “Le collezioni musicali nella Biblioteca Digitale Italiana di Internet culturale,” *Quaderni Estensi*, vol. IV, pp. 71–86, 2012, available at: http://www.quaderniestensi.beniculturali.it/.

[7] B. Dimić, B. Milosavljević, and D. Surla, “XML schema for UNIMARC and MARC 21,” *The Electronic Library*, vol. 28, no. 2, pp. 245–262, 2010.

[8] D. Holdsworth and D. M. Sergeant, “A blueprint for representation information in the OAIS model,” in *IEEE Symposium on Mass Storage Systems*, 2000, pp. 413–428.

[9] H. v. d. Sompel, M. L. Nelson, C. Lagoze, and S. Warner, “Resource harvesting within the OAI-PMH framework,” *D-Lib Magazine*, vol. 10, 2004.

[10] A. Baratè and L. A. Ludovico, “IEEE 1599 applications for entertainment and education,” in *Music Navigation with Symbols and Layers: Toward Content Browsing with IEEE 1599 XML Encoding*, D. Baggi and G. Haus, Eds. Wiley-IEEE Computer Society Press, 2013, pp. 115–132.

[11] D. Baggi and G. Haus, “IEEE 1599: Music encoding and interaction,” *Computer*, vol. 42, no. 3, pp. 84–87, 2009.

[12] A. Baratè and L. A. Ludovico, “Local and global semantic networks for the representation of music information,” *Journal of e-Learning and Knowledge Society*, vol. 12, no. 4, pp. 109–123, 2016.

[13] A. Baratè, G. Haus, L. A. Ludovico, and P. Perlasca, “Managing intellectual property in a music fruition environment - the IEEE 1599 case study,” *IEEE MultiMedia*, vol. 23, no. 2, pp. 84–94, 2016.

[14] L. A. Ludovico and G. R. Mangione, “An active e-book to foster self-regulation in music education,” *Interactive Technology and Smart Education*, vol. 11, no. 4, pp. 254–269, 2014.

[15] A. Baratè, G. Haus, L. A. Ludovico, and D. A. Mauro, “IEEE 1599 for live musical and theatrical performances,” *Journal of Multimedia*, vol. 7, no. 2, pp. 170–178, 2012.

MACAQUE – A TOOL FOR SPECTRAL PROCESSING AND TRANSCRIPTION

Georg Hajdu
Center for Microtonal Music and Multimedia (ZM4)
Hamburg University of Music and Theater (HfMT)
Harvestehuder Weg 12
20148 Hamburg

georg.hajdu@hfmt-hamburg.de

ABSTRACT

This paper describes *Macaque*, a tool for spectral processing and transcription, in development since 1996. *Macaque* was programmed in *Max* and, in 2013, embedded into the *MaxScore* ecosystem. Its GUI offers several choices for the processing and transcription of SDIF partial-track files into standard music notation. At the core of partial-track transcription is an algorithm capable of "attracting" partial tracks (and fragments thereof) into single staves, thereby performing an important aspect of "spectral orchestration."

1. INTRODUCTION

Macaque is a component of the *MaxScore* notation software package for Max [1] allowing the transcription of analysis data in the Sound Description Interchange File Format (SDIF) into standard music notation. It has a long history dating back to March 1996 when, during a ZKM residency in Karlsruhe, Germany, I tried to recreate the workflow I used for my doctoral work at UC Berkeley's Center for New Music and Audio Technologies (CNMAT). At CNMAT, I took advantage of the analysis component of the their additive synthesis tool (CAST) running on a Silicon Graphics Indigo computer [2][3]. In contrast, the software available to me in Karlsruhe consisted of an application called *Lemur* running on the classic Mac OS for partial-tracking analysis as well as *Finale* by Coda (now MakeMusic) for music notation. Lemur implemented the McAulay and Quatieri algorithm [4] capable of modeling non-harmonic and polyphonic sounds [5] and was further developed into *Loris*, an "Open Source sound modeling and processing software package based on the Reassigned Bandwidth-Enhanced Additive Sound Model" [6][1]. I used *Lemur* for partial-tracking analysis and a *Max* patch to translate the analysis data from binary into text format and, eventually, into a MIDI file. Once imported into *Finale*, the files were exported in *Enigma* format—*Finale's* file exchange format until it was superseded by MusicXML. The *Enigma* format (despite its name) allowed me to alter the appearance of my scores by changing the cryptic code in specific locations. To this aim, I developed a number of *Max* patches. For instance, in the first scene of the second act of my opera *Der Sprung – Beschreibung einer Oper* [7] I used the MIDI velocity information of the transcribed note events to alter the size of their note heads so that the sight-reading musicians had instantaneous visual feedback pertaining to the dynamics of the music to be performed. In other instances, I have used a technique called "velocoding" to encode microtonal pitch deviation in eighth-tone resolution into the velocity part of a MIDI note-on message to modify the Enigma file in such ways that the resulting score displayed the corresponding pitch alterations. Another early example of using *Macaque* is my piece *Herzstück* for two player pianos from 1999 which premiered at the Cologne Triennale in 2000. This piece was written for two of Jürgen Hocker's instruments which he also used to tour Conlon Nancarrow's compositions for player piano [8]. These instruments had been retrofitted with a mechanism allowing them to be controlled via MIDI. In my piece, the two pianos were to "speak" the eponymous comical dialog by Heiner Müller, once dubbed the world's shortest theatre piece, with its length well below a minute. I used an audio recording by the *Berliner Ensemble* and also translated the background noises such as the frantic applause at the end.

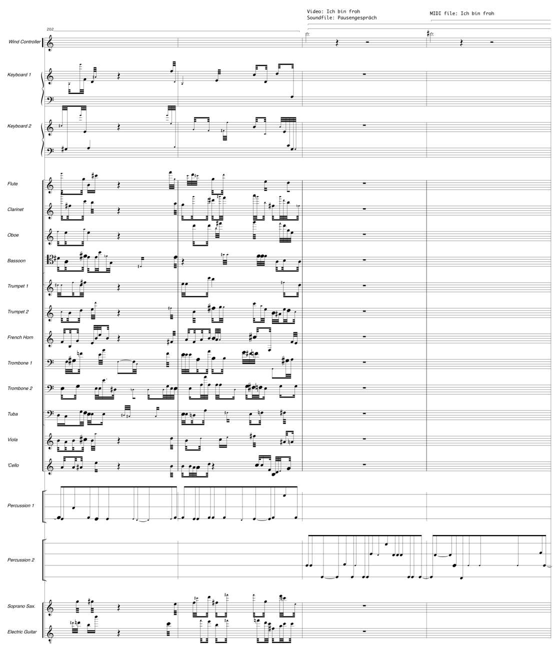


Figure 1. Excerpt from the first scene of the second act from the author's opera *Der Sprung – Beschreibung einer Oper*.

[1] *Macaque* was named after another simian whose name connotes the name of the platform it was developed on as well as the tongue-in-cheek reference to "aping" real sounds with additive or instrumental resynthesis.

In the early 2000s, *Macaque* went through several steps until it reached its current incarnation, among these were:

- Adaption of the SDIF format co-developed by IRCAM and CNMAT
- Switch to SPEAR and AudioSculpt as a source for SDIF files
- Implementation of spectral transforms and time stretching
- Development of a collapsible GUI with three separate panes
- Integration into the MaxScore ecosystem

2. SPECTRAL COMPOSITION AND ORCHESTRATION

At the core of *Macaque* is a technology intelligently assigning partial tracks to *event tracks* (see section 3.3). Partial-tracking analysis of complex sounds typically produces more tracks than an ensemble of musicians can handle. They typically either exceed the number of available musicians or the playable range of their instruments. Therefore, files generated with SPEAR should be prepared in advance. These preparations involve:

- Setting appropriate values in SPEAR's Sinusoidal Partials Analysis window (**Figure 2**) depending on the source type (instrumental sounds, music, noise, speech)
- Defining a cut-off frequency and removing partials outside the range with the Frequency Region Selection tool (typically 3000 Hz)
- Removing short partial tracks (typically <=0.2")
- Deleting soft partial tracks or "false" tracks consisting of noise
- Manually editing partial tracks where a signal has fused with noise

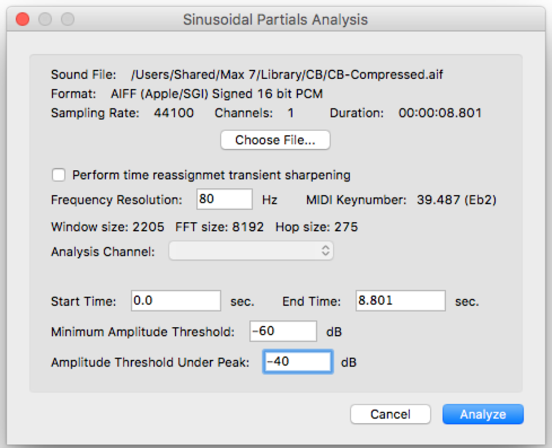


Figure 2. It is crucial to start with the right settings in the Sinusoidal Partials Analysis window before analyzing an audio file in SPEAR.

Still, this may not be enough to sufficiently reduce the number of tracks. In section 3.3, I will therefore describe an algorithm that "attracts" separate partial tracks into an instrumental staff and thereby performs, on a rudimentary level, a task which can be called *Spectral Orchestration*.

There have been a number of projects by other composers and developers tackling aspects of spectral orchestration. Those known to me include the works by the French spectralists, the software Clarence Barlow's developed for his piece *Am Januar am Nil* (1980) in which nonsense sentences are "spoken" by an ensemble, the piece *Speakings* by Jonathan Harvey for which the Matlab-based software *Orchidée* was developed at IRCAM [9]; more on this in a paper by Aurélien Antoine and Eduardo R. Miranda [10]. Other projects include the *soundalikes* by Michael Iber, the text compositions by Peter Ablinger, as well as OpenMusic [11] and the bach/cage libraries [12] capable of converting SDIF files into music notation.

3. WORK FLOW

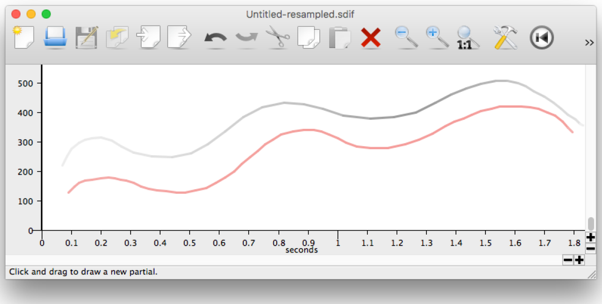


Figure 3. An example of partial tracks in SPEAR. These tracks were actually drawn by hand.

Including *Macaque* into the *MaxScore* ecosystem has simplified the workflow to a great extent and allowed me to work mainly in the Max environment. The following sections will give an overview of the crucial steps from SDIF import to score generation.

3.1 Importing from Spear

SDIF files such as the one displayed in **Figure 3** should be exported from SPEAR as "SDIF 1TRC – Exact Interpolated". *Macaque* can be conveniently accessed from within a patch called *Macaque Environment*, which is part of the *MaxScore* ecosystem and also comprises an instance of the *MaxScore* editor, a *Macaque* sound file recorder (for resynthesized SDIF files) and two modules for microtonal and multitimbral playback.

3.2 The GUI

Macaque sports three panes and four tabs for the top pane (**Figure 4**). The default view displays (i) the partial tracks in the top pane with the Transcribe button underneath (triggering the transcription of partial tracks into notation), (ii) the spectral content of a vertical time slice (spectral frame) in eighth-tone notation in the central pane as well as (iii) curves for centroid (green) and sum of amplitudes (black) in the bottom pane.

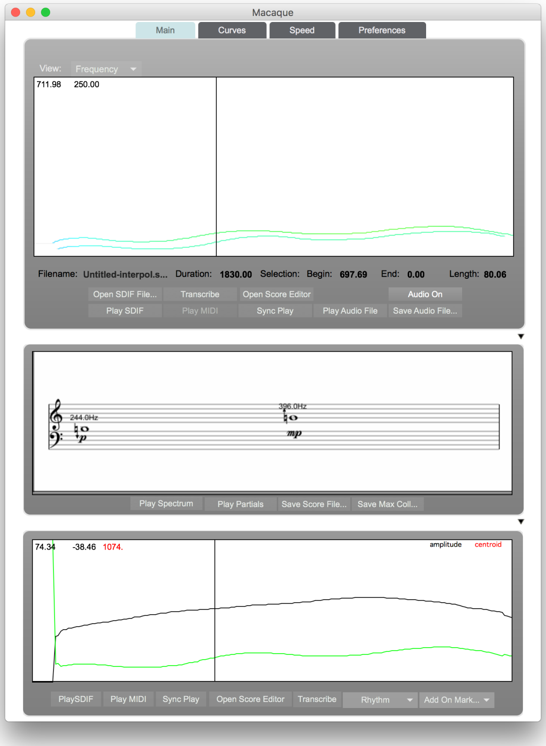


Figure 4. The *Macaque* GUI with its collapsible panes

These curves serve as the basis for event detection and markup, as we will see in section 3.5. A second Transcribe button triggers event transcription according to the markers created by the user.

The other three views of the top pane display:

- Break-point functions for spectral transforms (**Figure 5**)
- A tempo curve for time stretching/compression and
- A preferences pane with over 15 parameters affecting the outcome of the transcription (**Figure 6**)

Figure 5. Break-point functions for spectral transforms

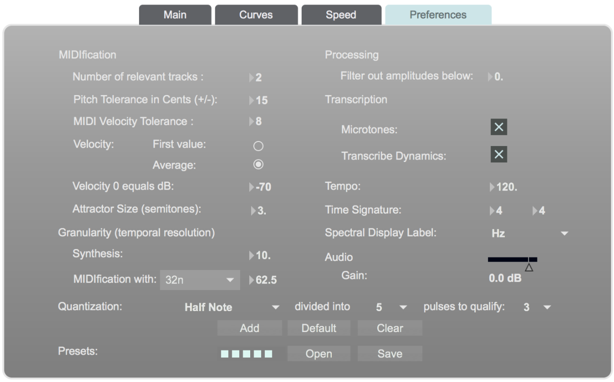


Figure 6. Preference pane with parameters affecting the outcome of the transcription

3.3 Partial-track transcription

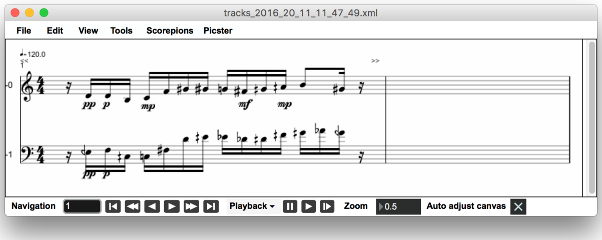


Figure 7. Transcription of the partial tracks from **Figure 3**

Macaque relies to a great extent on the CNMAT sdif objects (sdif-buffer, sdif-info, sdif-ranges, sdif-tuples) [11]. Upon opening an SDIF file in 1TRC format and loading stream number 0 (higher stream numbers are currently not supported) into the sdif-buffer, relevant information about the file is extracted and the spectral content displayed in the top pane by reading the data from the SDIF matrix contained in the buffer.

3.3.1 MIDIfication

Pressing the transcribe button will now pass the spectral data to the transcriber, at the time interval defined as MIDIfication in the Granularity preference section. This interval is calculated by taking current meter, tempo and beat subdivision settings into consideration (**Figure 7**). Note that the quantizer offers another *beat subdivision scheme* [14] which can either be aligned with the MIDIfication interval or not. When aligned the subdivision is regular, if not the subdivision is irregular and notes may be lumped together such as in **Figure 8**.

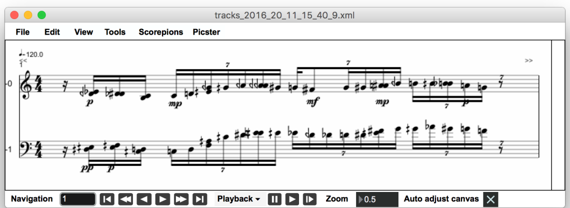


Figure 8. Transcription of the same partial tracks with misaligned MIDIfication and beat subdivision scheme settings. This may or may not be a desired effect.

After applying the spectral transforms (see section 3.3.3) the partial tracks are resampled according to their index in a 32-bit Jitter matrix. Each track is converted and analyzed according to Pitch and MIDI Velocity tolerance thresholds, i.e. the analysis looks for leaps in the resampled values exceeding a given threshold.

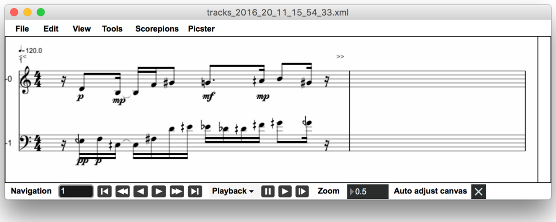


Figure 9. Transcription of the same partial tracks with vastly different Pitch and MIDI Velocity tolerance thresholds.

If a value is greater than the threshold value, a new event is assumed and the events collected in a Max coll (see **Table 1**). Each track now consists of a track velocity value (based either on the first collected amplitude value or an average of all amplitude values), an average track pitch value and four values for each event consisting of time tag (in MIDIfication intervals), event frequency, event velocity and duration (in MIDIfication intervals).

#	Events in track
0	0.14899 65.489636 1 62.14 31. 2 3 59.07 48. 2 5 64.84 53. 1 6 68.16 58. 2 8 67.11 65. 3 11 69.62 61. 1 12 71.21 51. 2 14 67.95 51. 1;
1	0.050958 59.706389 1 51.49 25. 1 2 53.14 33. 1 3 48.67 38. 2 5 53.91 38. 1 6 61.9 40. 1 7 64.54 39. 1 8 63.05 40. 1 9 61.18 42. 1 10 62.37 42. 1 11 65.33 41. 1 12 67.69 41. 2 14 66.28 41. 1;

Table 1. Event collection. Each track is represented by average velocity, average pitch and a sequence of four values denoting time tag, frequency, velocity and duration for each individual event.

3.3.2 Event Attractor

These data serve as the basis for an algorithm assigning these events to event tracks. It works as follows: The track velocity serves as a measure for its relevance; the louder the track the more relevant. All tracks are indexed according to this measure. For the first partial track (the most relevant track), all events are written to another Jitter matrix and now serve as an *attractor* to events which exist in the other tracks. If the events of the next track are close enough in pitch (defined by *Attractor Size* in the *Preference* pane) and can be inserted into empty regions of the current event track, they will be written to this track, otherwise a new event track will be created (Figures 10-12). This process is iterated until all events have either been assigned to event tracks or discarded, the maximum count being 32.

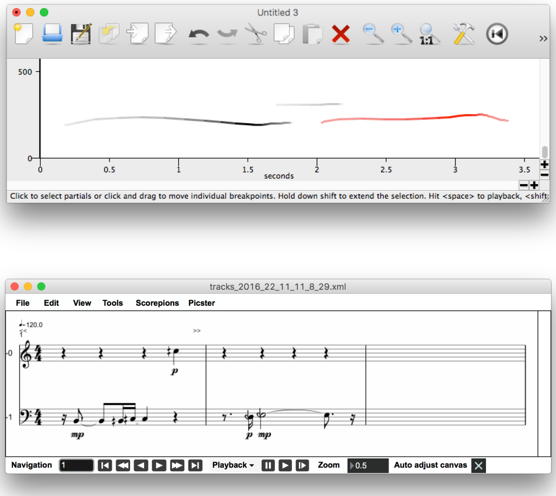


Figure 10. Transcription of overlapping partial tracks yields separate staves.

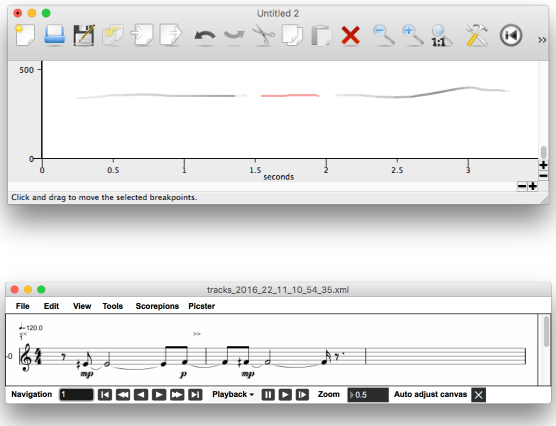


Figure 11. Transcription of consecutive partial tracks within attractor range yields one staff.

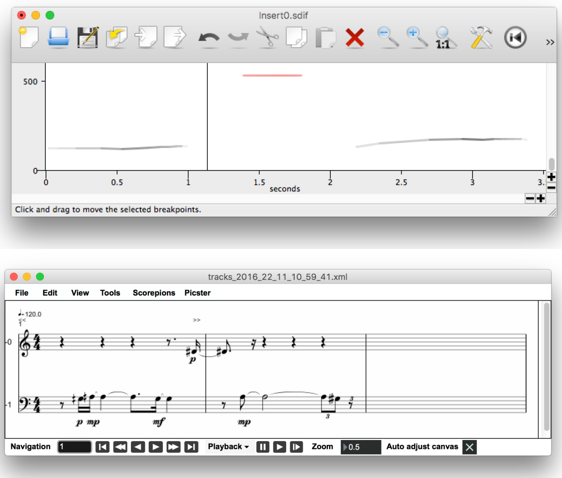


Figure 12. Transcription of consecutive partial tracks outside attractor range yields two staves.

The next steps involve sorting event tracks according to their average pitch as well as converting time tags and durations into their respective values in seconds. This is where time stretching and compression is applied (**Figure**

14). Finally, these values are fed into the *MaxScore* transcriber and displayed in standard notation.
Once transcribed the original SDIF file and its companion score file can be played back in sync by pressing the "Sync Play" button.

3.3.3 Spectral transforms

Macaque can apply time-variant transforms to spectral data. These transforms can be set by changing breakpoint functions (BPFs) for *amplitude*, *trajectory*, *spectral stretch*, *reference frequency* and *transposition*. They are also being applied to the playback of the SDIF file. I used Emmanuel Jourdan's *ej.function.js* JavaScript object capable of drawing multiple BPFs on top of each other and sharing its curves with an efficient Java object called *ej.fplay* for real-time processing.
While the terms *amplitude* and *transposition* don't need further elucidation, I'd like to explain the function and meaning of *trajectory* and *spectral stretching*. *Trajectory* refers to the path playback and transcriber take through the SDIF file. A straight upward line causes the sample to be played regularly, i.e. forward, a straight downward line causes the sample to be played backwards. By using any number of break points, playback and transcription can be broken up into forward and backward segments. Spectral stretching is performed according to the formula given by Mathews and Pierce [15]. It requires the partial index (defined as the ratio between partial and reference frequency), a pseudo-octave (or stretch factor; 2 = no stretch) and reference frequency (or fundamental) as inputs.

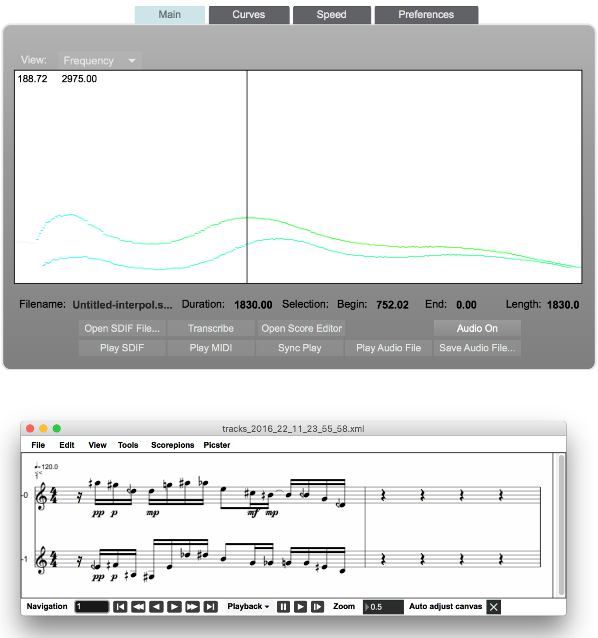


Figure 13. The transcription of the same file with spectral-stretching applied. The stretch factor is 2.95 at the beginning shrinking linearly to 1.68 over the length of the file.

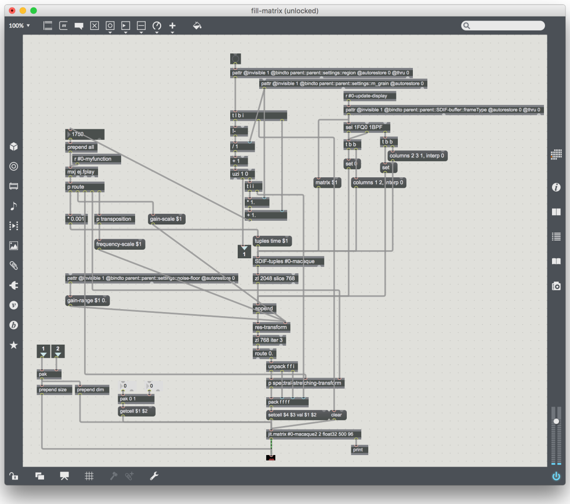


Figure 14. The subpatch in *Macaque* where spectral transforms are applied. Note the use of the *ej.fplay* object which shares the break-point functions of the *Curves* pane.

3.3.4 Tempo curve

Another editor can be used to warp time according to a time-variant tempo curve, i.e. portions of the sample can be sped up or slowed down. This tempo curve will then be applied to transcription. This is performed by calculating the integral under the tempo curve to obtain the values for duration and onset times of the note events to be used by the *MaxScore* transcriber.

3.3.5 Selection

Playback, transcription and transforms all scale to the selection made in the partial-track pane. This affords the user the possibility to specify select parts of the SDIF file.

3.3.6 Effects processing

Since all transforms are also applied to the resynthesis of the SDIF file, *Macaque* can also be used as an effects processor. Two of my compositions (see Table 2) have taken advantage of this capability.

3.4 Spectral frames

As mentioned before, the spectral slices pane displays the spectral content at the position of the play head. Its partials can be played back at once by holding the space bar or arpeggiated by pressing the "Play Partials" button. They can be saved as a *MaxScore* XML file or a Max coll. In addition, the notes can be copied and pasted into a *MaxScore* score for compositional or analytical purposes.

3.5 Event detection and transcription

The third pane is dedicated to event detection. Here, the term event is applied to the entire analysis file, in contrast to partial track transcription where "event" refers to the detection of significant amplitude and frequency changes within a single track. As automatic event detection poses some challenges [16], I have opted to (i) offer markers for manual event detection (for which the amplitude and centroid curves lend themselves as guides) or (ii) for the import of markers generated in *AudioSculpt*, whose algorithm performs better than the one employed in an earlier version of *Macaque*. Once an SDIF file has been marked up with on-markers and off-markers, the obtained temporal structure serves as the basis for transcription and further processing. Two adjacent on-markers delineate a time interval within which the spectral frame with the highest sum of amplitudes is searched for. From this frame, the following events can be derived:

- Temporal structure with x-ed note heads.
- f0 pitch, applying the harmonic histogram technique implemented in Mikhail Malt and Emmanuel Jourdan's zsa.fund object
- Lowest partial
- Most salient partial
- Centroid
- The nearest neighbor of the centroid, as the latter is typically not contained in the spectrum
- All partials as a chord (an amplitude threshold can be set in the preference pane to skim off softer partials)

While the markup should ideally follow sharp rises or drops in the amplitude and/or centroid curves, markers can also be set to apply arbitrary rhythms to the spectrum of a sounds (**Figure 15-17**).

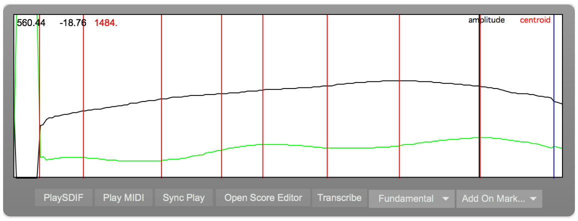


Figure 15. An (arbitrary) markup of the SDIF file from **Figure 3**

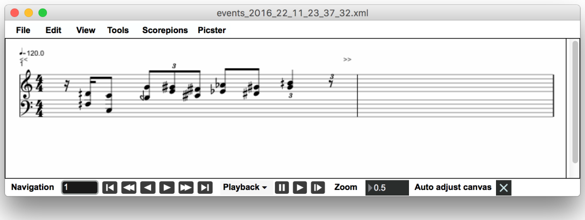


Figure 16. The transcription of the markup displaying the strongest partials within the delineated segments.

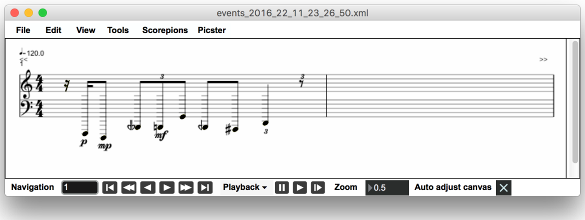


Figure 17. Same markup displaying f0 pitch.

4. COMPOSITIONS

After a hiatus of nearly 10 years during which I mainly focused on networked multimedia performance, I started to create spectral music again in 2009. Since then I have used *Macaque* in the following compositions (see http://georghajdu.de):

Composition	Year	Instrumentation	SRC	PT	SF	MU	ASUP	RSYN
Blueprint	2009	sax, egtr, db, pno, perc, elec	speech, noises	x	x	x	x	
Schwer... unheimlich schwer	2009	bcl, vla, pno, perc, elec	speech		x	x	x	
Swan Song	2011	vc, perc, elec	Beijing opera, noises	x		x	x	x
In ein anderes Blau	2012	sop, bfl, cbcl, vn, va, vc, db, perc, elec	music		x		x	x
noiwont	2014	19-tone trp, elec	speech	x		x	x	
aɪd laɪk tuː meɪk ə ʃɔːt 'steɪtmənt	2016	fl, cl, va, vc, pno, perc, elec	speech		x	x	x	

Table 2. Compositions by the author composed with the aid of *Macaque*. SRC = source material, PT = partial-track transcription, SF = spectral frame, MU = Markup and event transcription, ASUP = audio superimposition, RSYN = SDIF to audio resynthesis.

5. CONCLUSION AND OUTLOOK

Emerging from a situation in which I desperately needed to replace software I had relied on during my doctoral work at CNMAT, *Macaque* has become, over the years, a serious tool used by me, and others, for spectral analysis and composition. It has become fairly stable in its feature set for the past 4 years with development mainly focusing on bug fixes and support for 64-bit Max.
However, there are a few areas that are still worthwhile exploring:

- Transcription and notation of glissandi for events belonging to the same partial track
- Implementation of an efficient automatic event detection algorithm with a "rubber-band" tempo curve editor capable of taking tempo fluctuations and microtiming into consideration (see also [17])
- Implementation of a fast method for tempo curve integration
- Zooming
- Improvements of the GUI

Since the code base largely emerged before the release of Max 5, it is tempting to recreate the functionality of the partial-track transcriber and other components in the Max js object as well as improve control of the additive synthesis as a Max gen~ script.

Acknowledgments

I would like to thank the *Behörde für Forschung und Wissenschaft Hamburg* for supporting our research in the framework of its *Landesforschungsförderung.*

6. REFERENCES

[1] G. Hajdu, and N. Didkovsky, "MaxScore – Current State of the Art," in *Proceedings of the International Computer Music Conference*, Ljubljana, 2012, 156-162.

[2] A. Freed, X. Rodet and Ph. Depalle, "Synthesis and Control of Hundreds of Sinusoidal Partials on a Desktop Computer without Custom Hardware," in *ICSPAT (International Conference on Signal Processing Applications & Technology*, 1992, San José, United States. 1992

[3] http://cnmat.org/CAST/

[4] R.J. McAulay and T.F. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," in *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. ASSP-34, no. 4, 1986, 744-754.

[5] K. Fitz and L. Haken, "Lemur: A bandwidth-enhanced sinusoidal modeling system," in *The Journal of the Acoustical Society of America,* vol. 103, 1998, 2756.

[6] http://www.hakenaudio.com/Loris/

[7] G. Hajdu, "Research and Technology in the Opera Der Sprung," in *Nova Acta Leopoldina*, vol. 92, no. 341, 2005, 63-89.

[8] J. Hocker, *Faszination Player Piano - Das Selbstspielende Klavier von den Anfängen bis zur Gegenwart. Kapitel 15 - Conlon Nancarrow und die Renaissance des Player Pianos,* Edition Bochinsky, 2009, 238-267.

[9] G. Nouno, A. Cont, G. Carpentier and J. Harvey, "Making an Orchestra Speak," in *Proceedings of the Sound and Music Computing Conference (SMC),* Porto, 2009, 277-282.

[10] A. Antoine and E. R. Miranda, "Towards Intelligent Orchestration Systems," in *Proceedings of the 11th International Symposium on Computer Music Multidisciplinary Research (CMMR): Music, Mind, and Embodiment*, Plymouth University, UK, 2005.

[11] J. Bresson and C. Agon, "SDIF sound description data representation and manipulation in computer assisted composition," in *Proceedings of the International Computer Music Conference*, Miami, USA, 2004, 520–527.

[12] A. Agostini, E. Daubresse and D. Ghisi, "cage: a high-level library for real-time computer-aided composition," in *Proceedings of the Joint ICMC and SMC conference,* Athens, 2014, 308-313.

[13] M. Wright, R. Dudas, S. Khoury, R. Wang and D. Zicarelli, "Supporting the Sound Description Interchange Format in the Max/MSP Environment," in *Proceedings of the International Computer Music Conference*, Beijing, 1999, 182-185.

[14] N. Didkovsky, "Java Music Specification Language, v103 update," in *Proceedings of the International Computer Music Conference*, Miami, 2004, 742-745.

[15] M. Mathews and J.R. Pierce, "Harmony and Nonharmonic Partials," in *The Journal of the Acoustical Society of America*, vol. 68, 1980, 1252.

[16] N. Collins, "A Comparison of Sound Onset Detection Algorithms with Emphasis on Psychoacoustically Motivated Detection Functions," in *Proceedings of AES118 Convention*, 2005.

[17] http://fab.cba.mit.edu/classes/864.05/people/lieb/lev.html

Webpages all accessed on November 23, 2016.

A MACHINE LEARNING FRAMEWORK FOR THE CATEGORIZATION OF ELEMENTS IN IMAGES OF MUSICAL DOCUMENTS

Jorge Calvo-Zaragoza
Software and Computing Systems
University of Alicante
Alicante, Spain
jcalvo@dlsi.ua.es

Gabriel Vigliensoni, Ichiro Fujinaga
Schulich School of Music
McGill University
Montréal, Canada
{gabriel,ich}@music.mcgill.ca

ABSTRACT

Musical documents may contain heterogeneous information such as music symbols, text, staff lines, ornaments, annotations, and editorial data. Before any attempt at automatically recognizing the information on scores, it is usually necessary to detect and classify each constituent layer of information into different categories. The greatest obstacle of this classification process is the high heterogeneity among music collections, which makes it difficult to propose methods that can be generalizable to a broad range of sources. In this paper we propose a novel machine learning framework that focuses on extracting the different layers within musical documents by categorizing the image at pixel level. The main advantage of our approach is that it can be used regardless of the type of document provided, as long as training data is available. We illustrate some of the capabilities of the framework by showing examples of common tasks that are frequently performed on images of musical documents, such as binarization, staff-line removal, symbol isolation, and complete layout analysis. All these are tasks for which our approach has shown promising performance. We believe our framework will allow the development of generalizable and scalable automatic music recognition systems, thus facilitating the creation of large-scale browsable and searchable repositories of music documents.

1. INTRODUCTION

Optical Music Recognition (OMR) is the branch of artificial intelligence focused on automatically recognizing the content of a musical score from the optical scan of its source. In comparison to similar tasks such as text recognition, this process can be quite difficult given the complexity of music notation and the wealth of information contained in these documents. In addition to the musical notes that are usually overlaid on the staff lines, music scores may also contain several types of heterogeneous information such as alterations, lyrics, decorations, or bibliographic information about the piece. Therefore, before any attempt of automatic recognition, it is important to detect and classify these elements into their corresponding categories.

In addition to the tasks of symbol recognition and classification, there are other OMR preprocessing operations that are less well known. For example, a common first step in OMR workflows is binarization. This process consists in separating the background (i.e., the superfluous part of the image) from the foreground (i.e., the relevant content), and is usually considered the starting point for the subsequent OMR steps. A typical task that follows the binarization process is the detection and removal of staff lines. Although these lines are necessary for human readability and music interpretation, most OMR workflows are based on detecting and removing the staff lines before doing the classification of the remaining elements in the score.

OMR preprocessing is a complex step. In the past few years, many researchers have proposed OMR algorithms, workflows, and systems that deal with specific tasks on music documents, such as binarization [1], staff-lines detection [2], frontispiece delimitation [3], measure recognition [4], extraction of lyrics [5], and page border removal [6]. These approaches were all based on heuristical rules tailored to the music corpus at hand and achieved varying performance. Music documents have a high level of heterogeneity and exhibit many sources of variability, such as image degradation, bleed-through, different notation types, handwritten styles, or ink differences, among others. Therefore, if OMR systems are implemented by taking advantage of specific characteristics of the documents, different algorithms may be needed when working with sources of different type. As a result, the implementation of these systems will lack of generalizability and may be one of the factors hindering the progress of OMR technology.

In order to ameliorate this situation, we propose a generalized framework that allows detecting the different layers (i.e., background, staves, music symbols, lyrics, and so on) from the image of a music score, regardless of the specific characteristics of the source document. Extending the idea initially proposed by Calvo-Zaragoza et al. [7] for detecting and removing staff lines by using machine learning, we propose an approach in which each pixel of the image is labeled according to the type of content it depicts.

In contrast to strategies based on heuristic image processing, the main advantage of using machine learning rests in its generalizability. While the former focuses on particular aspects of the scores—being therefore very difficult to adapt to other documents—techniques based on machine learning only need examples of the new type of documents to generate a different model. In some cases, it is even possible to reuse already trained models in documents of similar nature, but with a different type or style, by using Transfer Learning techniques [8].

Until a few years ago, the main disadvantage of using machine learning systems was that they did not achieve good results for image recognition tasks. However, since the rise of Deep Learning [9], Convolutional Neural Networks (CNN) have completely changed the scenario, outperforming traditional techniques in these tasks [10].

The rest of this paper is structured as follows: in Section 2 we detail the proposed unified framework and the rationale behind it. In Section 3 we show examples of tasks that can be successfully performed with the proposed framework. Finally, in Section 4 we summarize the core ideas of our method and gives some hints about future work.

2. DESCRIPTION OF THE FRAMEWORK

The framework we propose is based on the categorization of each pixel of interest within the input image with the label that illustrates to which information layer it belongs. To perform this task, we make use of the supervised learning paradigm [11]. That is, it is assumed that there will be enough representative examples of each type of information layer to be able to create a model to categorize new, unseen examples. Three elements are therefore essential for implementing this approach: (i) a feature set for each pixel, (ii) a classification algorithm, and (iii) training data.

2.1 Feature set

The feature set must characterize appropriately the pixel to be classified. We assume that the region of pixels around a specific pixel contains enough discriminating information to classify it with success. In other words, we hypothesize that a pixel can be correctly categorized by using the local information surrounding it. For example, whereas areas with staff lines may usually indicate zones where music notation is, areas without staff may indicate that other content, such as ornaments or lyrics, may be present. Text and decorations are similar in the local sense, but different ink type, color, or pen trace may have been used. Our approach exploits these local features to correctly distinguish the categories of the different elements within a musical document.

Figure 1 shows three examples of features sets for different pixels of an image. The pixel to be classified is located at the center of each window. Note that the size of the neighborhood (i.e., the size of the window) is a parameter to be tuned empirically, as the performance is highly related to this value [7].

Depending on the task, it might be advisable to increase the size of the window so that the features are discriminative enough. For example, with a small window it is possible that the feature set of a *text* sample would not be very different from those of *musical symbol*. However, increasing the size of the window too much may lead to

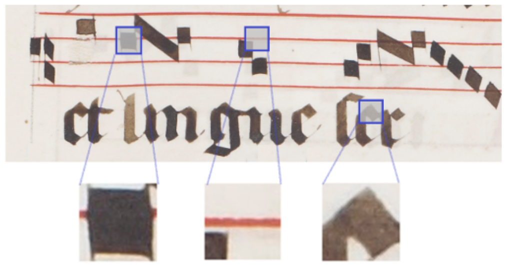



Figure 1. Example of feature sets from three regions of interest (i.e., music symbols, staff lines, and text). The pixel to be classified is located at the center of each window.

an increase in the complexity of the problem, which could make the CNN not learn the task correctly. In addition, as the size of the feature set increases, a more computational time is needed.

2.2 Classification algorithm

In our framework, the classification process is carried out by means of Deep Learning. Recently, Deep Neural Networks have shown a remarkable leap of performance in the field of machine learning. Specifically, CNN have been applied with great success for the detection, segmentation, and recognition of objects and regions in images, approaching human performance on some of these tasks [10].

These neural networks are composed of a series of filters (i.e., convolutions) that allow obtaining several representations of the input image. These filters are applied in a hierarchy of layers, each of which represent different levels of abstraction: whereas filters of the first layers enhance details of the image, filters of the last layers detect high-level entities [12]. The key is that these filters are not fixed but learned through a gradient descent optimization algorithm called back-propagation [13]. The configuration and organization of the network hierarchy (usually referred to as *topology*) has to be designed or chosen by the researcher.

Since collections of music documents are a rich source of highly heterogeneous information—usually more complex than other types of documents—developing a unified framework for OMR with a classification algorithm based on CNN is promising.

2.3 Training data

The last component to be considered in our framework is training data, which is dependent on the specific type of task to be performed. For example, it is likely that data needed to train a model to detect staff lines is different from data needed to discriminate among other items, such as musical symbols or text. Either way, the need of training data is the main drawback for the proposed framework, since it has to be created by manually labeling examples of all regions of interest in the document.

It is worth mentioning that we do not consider the possibility that a pixel belongs to more than one class at a time. We believe that from the point of view of an OMR system, in most cases there is just a single label that is truly relevant. For example, pixels belonging to a musical symbol that are on a staff line should be considered as part of the former. But if needed, new categories for possible overlapping elements could be added, allowing the system to learn these categories as well.

3. EXAMPLES

In the following we present a number of examples of tasks in the classification of elements within musical documents. We made use of a CNN topology consisting of three convolutional layers. Although this might not be the best topology for the problems at hand, it is illustrative of the classification-based approach we propose. The window size of the feature set was specifically tuned for each example by means of informal testing.

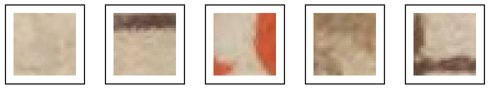
The approach presented in this paper is directly applicable to any type of document no matter the type of notation and the style of the score, as long as enough training data is given to the network. In fact, different sources were considered for each of the examples in order to show how generalizable is our approach.

3.1 Binarization

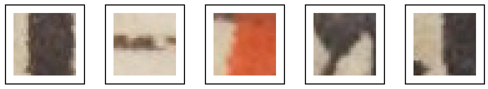
Binarization plays an important role in document analysis systems. This process is usually performed in the first stages of OMR systems and affects all subsequent stages. Therefore, it is crucial that binarization behaves in a robust way. Traditional binarization methods, however, have not shown consistent performance on music documents of different type. The degradation of music sources is one of the reasons for the unreliability of this process, but also great diversity in music notation is another obstacle [14].

The training data for this binarization example was composed of two manually labeled folios from Einsiedeln, Stiftsbibliothek, Codex 611(89). This manuscript is dated from 1314 and presents areas with severe bleed-through that may mislead standard binarization algorithms. From this labeled data, we selected the two layers of pixels that were labeled as *background* and *foreground*. We took random pixels from each layer and created a window of 25×25 pixels to be used as input feature for each pixel. We assumed that local information would be discriminative enough to classify correctly the center pixel. Figure 2 shows examples of features from both classes.

Once the CNN was trained with this data, it was able to distinguish between *background* and *foreground* pixels. As an illustrative example, Fig. 3 shows the binarization of a portion of a new document not seen during training that was classified pixel by pixel by the trained network. In spite of some spurious points that were misclassified, the network was able to achieve a remarkable performance for the binarization task.

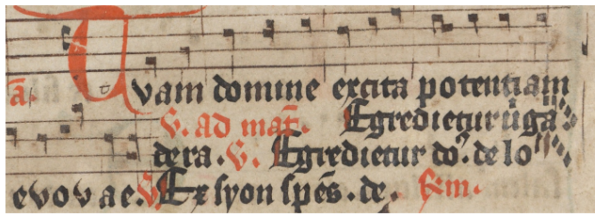


(a) Samples of *background* class

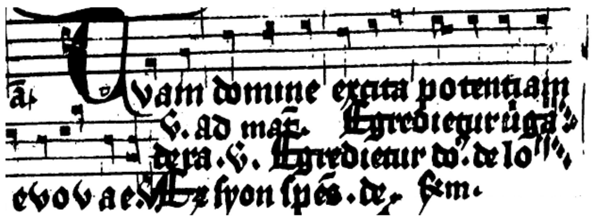


(b) Samples of *foreground* class

Figure 2. Training examples of both *background* and *foreground* classes. Each window has the pixel to be labeled at the center and also the local information to discriminate the class of the center pixel.



(a) Original input score portion



(b) Binarization of the input score

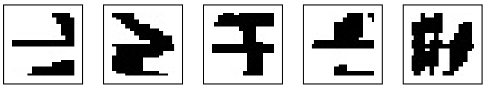
Figure 3. Example of binarization task performance achieved with our framework. The image was not part of the training set.

3.2 Staff-lines detection and removal

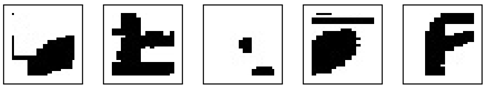
The detection and removal of staff lines follow the binarization step in most OMR workflows. Despite being necessary for musical readability, staff lines complicate the automatic detection, segmentation, and classification of symbols because they usually interconnect the symbols, thus not allowing their isolation.

Traditional methods for the staff-lines removal task consider a binary image as input because it helps to reduce the complexity of the problem. In addition, binarization is mandatory for applying processes based on morphological operators, histogram analysis, or connected components. The binary nature of modern music scores (i.e., blank ink on white paper) have justified somewhat this workflow.

In this example, we show how the removal of staff lines from binary images can be performed successfully with our framework. We trained the network with a dataset that provided enough information to distinguish between pixels that belong to *staff* or *symbol* classes. In this case, we took advantage of the CVC-Muscima database [15] because it was a dataset especially designed for the evaluation of staff-lines removal tasks and contains handwritten common modern notation scores with and without staves. Figure 4 shows windows of pixels belonging to both classes.

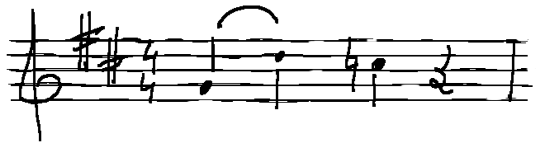


(a) Samples of *staff* class



(b) Samples of *symbol* class

Figure 4. Training data examples from *staff* and *symbol* classes.

We trained the CNN with enough data examples of the two classes, and then the network was able to detect and remove the staff lines accurately, as shown in Fig. 5.

(a) Example of input piece of score

(b) Input score after staff removal

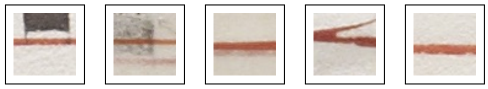
Figure 5. Staff-lines removal task with binary images achieved with our framework.

3.3 Symbol isolation on color images

As introduced above, traditional methods for staff-line detection require a binary image as input. Since binarization processes are highly sensitive to conditions of the documents such as irregular lighting, image skewing, inkblots, or paper degradation, the performance of the symbol isolation task depends largely on the previous steps of binarization and staff-line removal. Fortunately, if we detect both background and staff-lines at the same time, the approach we propose in this paper enables complete symbol isolation in just one step. As a result, for this task there are three possible categories to tag a pixel: *background*, *staff*, or *symbol*. The latter included both music symbols and text characters.

In order to demonstrate the adaptability of our framework, we decided to try a new set of musical documents, and so we trained the network with pixel samples from three full pages of the Salzinnes Antiphonal (CDM-Hsmu M2149.14) manuscript. Figure 6 shows examples of features for each category. A window size of 29 $\times$ 29 pixels was considered for this task.

(a) Samples of *background* class

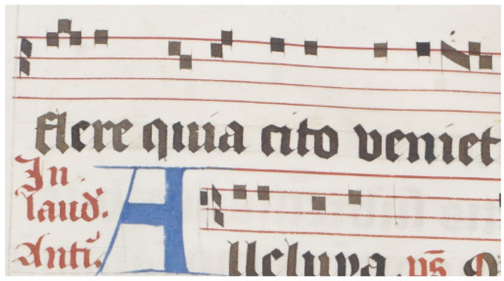


(b) Samples of *staff* class

(c) Samples of *text* class

Figure 6. Examples of pixel windows from *background*, *staff*, and *symbol* classes.

After training the network, our approach was able to classify pixels belonging to the three different categories, as shown in Figure 7. The result is accurate but the detected staff lines are thicker than the original ones, possibly implying that the approach is over-sensitive in the local sense. The most plausible explanation is that the CNN does not notice too much difference amongst adjacent pixels, since the features are practically the same. This means that a pixel that is not on a staff line, but close to it, may be detected as a staff-line pixel by the network.



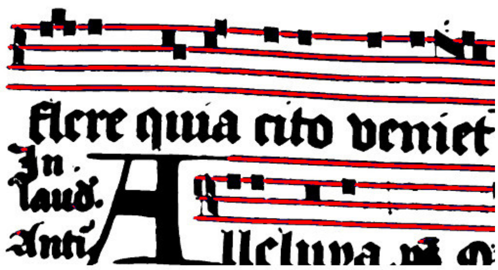
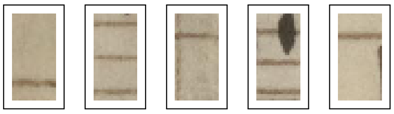
(a) Original input score portion

(b) Input score after staff-lines detection

Figure 7. Example of staff-lines detection on color images process achieved with our framework. Each layer considered is highlighted in a different color.

3.4 Music and text separation

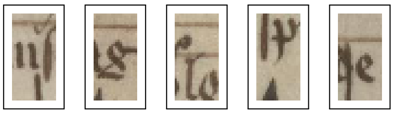
Music symbols and text are important sources of information in music documents. Due to their different nature, text and music are processed independently, with specialized automatic recognition algorithms. The proper separation of these two layers of information is a key aspect in the transcription of the whole document. We will show how our approach performs this classification task with ease.

In order to test the generalizability of our framework, we tested this task on a different music score, namely the GB-AR York Antiphonal manuscript. We manually classified pixels from one page into three different categories: *background*, *music*, and *text*. Fig. 8 shows a series of windows from each of these classes. Instead of a square window, preliminary experiments showed that a better perfomance was achieved with a rectangular window of 40×20 pixels.



(a) Samples of *background* class

(b) Samples of *music* class



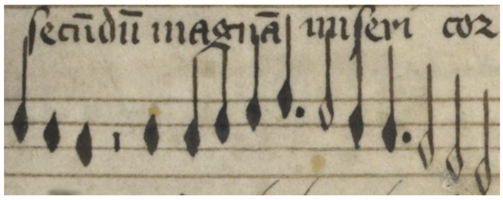
(c) Samples of *text* class

Figure 8. Examples of patches from *background*, *music*, and *text* classes.

Analogously to previous tasks, the CNN trained with these examples was able to produce accurate results, as shown in Figure 9. The framework achieved good performance even with those pixels where text and music symbols are overlapped. Nevertheless, as some pixels that belong to lyrics were erroneously classified as music, it is clear that the performance of this task is still not perfect.

This example highlights the strength of the framework we propose. It does not only separate text and music but it categorizes pixels at the pixel level—unlike previous approaches to this task that are devoted to just detecting zones or blocks of each type of information. Therefore, subsequent algorithms will not have to be in charge of performing the segmentation of the symbols within these blocks, since the specific pixels of interest are already detected.

3.5 Complete layout analysis

Typically, music scores contain much more information than just music symbols. This information includes titles, ornaments, lyrics, annotations, as well as unwanted artifacts such as ink bleed-through or ink blots. Therefore, a unified framework for complete document analysis of music documents should be able to identify and classify each of these categories within an input image. The framework we present in this paper is directly applicable to perform

(a) Example of input piece of score

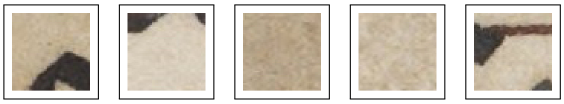
(b) Music and text separation in the input score

Figure 9. Detail of example of music and text separation using color images. Pixels classified as lyrics were labeled in red and symbols in black.

this task because it only needs enough training data and an appropriate window size surrounding each pixel.

Since the Einsiedeln manuscript contained several layers of interest within each page, such as music symbols, text, and ornamental letters, we tested a complete layout analysis in this manuscript. In this case, the data needed to be more discriminative and so we selected a window size of 51×51. As mentioned above, the specific size of the windows was chosen by performing preliminary experiments. What is important to remark in this case is that the window size needed to be larger than for the previous tasks because otherwise it would have been difficult to distinguish all categories. Also, since there were more categories, a larger amount of training data was required. Consequently, nine pages of the manuscript were manually labeled by categorizing their pixels into five different classes, namely *background*, *neume*, *text*, *staff*, and *decoration*. As a reference, the person in charge of building the training data required about 30 hours per page. Figure 10 shows a few examples of features extracted from this data, which were used to train the CNN.

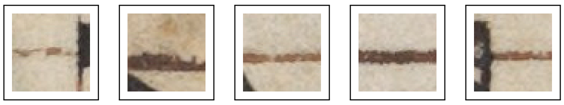
Figure 11 shows an example of the categorization achieved by our framework. It can be seen that the result was not optimal, especially in the case of distinguishing between music symbols and text. Given the proximity of music and text, the feature windows for both categories were similar. Nevertheless, this example shows that a complete layout analysis is feasible, regardless of the categories to be considered, as long as training data is available and the feature window size is tuned accordingly. As mentioned at the beginning of this section, our intention was not to achieve the best classification results, but to determine how the framework may be applied in a different number of tasks and music documents. Further efforts on the parameterization of the classifier scheme (i.e., CNN topology, training data, and features) need to be carried out to achieve a better performance.



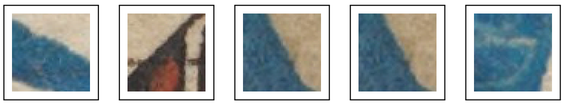
(a) Samples of *background* class

(b) Samples of *neume* class

(c) Samples of *text* class



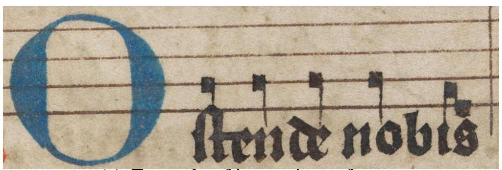
(d) Samples of *staff* class



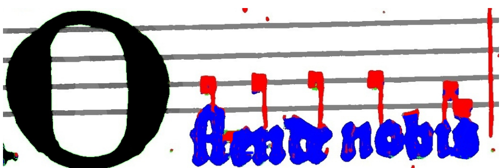
(e) Samples of *decoration* class

Figure 10. Examples of window patches from all the categories considered for the complete layout analysis task.

Once all the different elements within the documents have been grouped into the corresponding categories, music symbols can be classified, text can be processed by Optical Character Recognition applications, and the positions of the staff lines and their corresponding clefs can be used to determine the pitch of notes. In addition, ornamental letters can be either removed to not disturb recognition algorithms or kept for extracting their meaning. As a side benefit, the background has been detected conveniently, helping to reduce the complexity of the recognition tasks.



(a) Example of input piece of score



(b) Input score after layout analysis

Figure 11. Detail of the complete layout analysis achieved by our framework on a previously unseen score. Each layer considered is highlighted in a different color.

4. CONCLUSIONS

In this paper we presented a unified framework for categorizing information contained in digitized images of music documents. Unlike previously proposed approaches for OMR tasks, our work presents a highly generalizable and scalable method that allows performing any task of image recognition in any kind of musical document.

Our system labels individual pixels of the image depending on the information they contain. To do so, the system uses machine learning techniques, namely CNN, to learn from examples of each category to be classified.

We showed different tasks that can be performed with our framework, such as document binarization, staff-lines removal in binary and color images, music symbols and text separation, and complete layout analysis. All these tasks can be solved directly by just changing the training data provided to the framework and tuning the window size considered as feature set.

We are aware that the categorization of every pixel and element in music documents is only a part of the whole OMR problem. However, we believe that the unified framework presented in this paper will allow the development of generalizable and scalable OMR systems, thereby enabling a breakthrough towards large-scale automatic recognition of heterogeneous music documents.

As future work, efforts should be devoted to overcoming the problem of getting enough data to train the CNN. For the examples showed above, training data was obtained manually. Since this may be too costly if needed for each new kind of document, a more efficient process must be pursued. For instance, labeled documents depicting different conditions—such as scale, deformations, and so on—could be generated synthetically in order to get representative examples of each type. The use of adaptive techniques for Domain Adaptation or Transfer Learning is another way to deal with this issue [16]. Furthermore, it could be interesting to consider an incremental interactive framework in which the user does not have to label every single pixel of the image but only those erroneously labeled by a base classifier [17].

Acknowledgments

This work was partially supported by the Social Sciences and Humanities Research Council of Canada and the Spanish Ministerio de Educación, Cultura y Deporte through a FPU Fellowship (Ref. AP2012–0939). Special thanks to Vi-An Tran for manually labeling the different layers in all the manuscripts used for this research.

5. REFERENCES

[1] T. Pinto, A. Rebelo, G. A. Giraldi, and J. S. Cardoso, "Music score binarization based on domain knowledge," in *Proceedings of the 5th Iberian Conference on Pattern Recognition and Image Analysis*, 2011, pp. 700–8.

[2] C. Dalitz, M. Droettboom, B. Pranzas, and I. Fujinaga, "A comparative study of staff removal algorithms,"

IEEE Trans. Pattern Anal. Mach. Intell., vol. 30, no. 5, pp. 753–66, 2008.

[3] C. Segura, I. Barbancho, L. J. Tardón, and A. M. Barbancho, “Automatic search and delimitation of frontispieces in ancient scores,” in *18th European Signal Processing Conference*, 2010, pp. 254–8.

[4] G. Vigliensoni, G. Burlet, and I. Fujinaga, “Optical measure recognition in common music notation,” in *Proceedings of the 14th International Society for Music Information Retrieval Conference*, 2013, pp. 125–30.

[5] J. A. Burgoyne, Y. Ouyang, T. Himmelman, J. Devaney, L. Pugin, and I. Fujinaga, “Lyric extraction and recognition on digital images of early music sources,” in *Proceedings of the 10th International Society for Music Information Retrieval Conference*, 2009, pp. 723–7.

[6] Y. Ouyang, J. A. Burgoyne, L. Pugin, and I. Fujinaga, “A robust border detection algorithm with application to medieval music manuscripts,” in *Proceedings of the 2009 International Computer Music Conference*, 2009, pp. 101–4.

[7] J. Calvo-Zaragoza, L. Micó, and J. Oncina, “Music staff removal with supervised pixel classification,” *International Journal on Document Analysis and Recognition*, vol. 19, no. 3, pp. 211–9, 2016.

[8] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in *Advances in Neural Information Processing Systems*, 2014, pp. 3320–8.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 1097–1105.

[10] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–44, 2015.

[11] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York, NY: John Wiley & Sons, 2001.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *26th Annual Conference on Neural Information Processing Systems*, 2012, pp. 1106–1114.

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[14] J. A. Burgoyne, L. Pugin, G. Eustace, and I. Fujinaga, “A comparative survey of image binarisation algorithms for optical recognition on degraded musical sources.” in *Proceedings of the 8th International Society for Music Information Retrieval Conference*, 2007, pp. 509–12.

[15] A. Fornés, A. Dutta, A. Gordo, and J. Lladós, “CVC-Muscima: A ground truth of handwritten music score images for writer identification and staff removal,” *International Journal on Document Analysis and Recognition*, vol. 15, no. 3, pp. 243–51, 2012.

[16] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa, “Visual domain adaptation: A survey of recent advances,” *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 53–69, 2015.

[17] L. Pugin, J. A. Burgoyne, and I. Fujinaga, “MAP adaptation to improve optical music recognition of early music documents using Hidden Markov Models,” in *Proceedings of the 8th International Society for Music Information Retrieval Conference*, 2007, pp. 513–6.

A WEB INTERFACE FOR THE ANALYSIS AND PERFORMANCE OF ALEATORY MUSIC NOTATION

Adriano Baratè
LIM – Laboratorio di Informatica Musicale
Dipartimento di Informatica
Università degli Studi di Milano
adriano.barate@unimi.it

Luca A. Ludovico
LIM – Laboratorio di Informatica Musicale
Dipartimento di Informatica
Università degli Studi di Milano
luca.ludovico@unimi.it

ABSTRACT

Black and White n.2 is a collection of 120 exercises for keyboard instrument(s) written by the Italian composer Franco Donatoni. Conceived as aleatory music, this composition adopts a non-conventional way to encode the score where some parameters are fixed and others are left to chance. In this work, we will describe a Web-based framework that, after inserting user-defined scores in Donatoni's notation, is able to automatically produce score versions compatible with the composer's constraints and executable by a human player. This application produces modern staff notation and can perform it via the Web Audio API. The goal is on one side to revive the interest towards aleatory music literature, and Donatoni's repertoire in particular, and on the other to investigate the compositional and computational process that originate a given score out of many aleatory variants.

1. INTRODUCTION

In order to define *non-conventional music notation*, often the reference is the so-called Common Western Notation (CWN), namely the archetypical notation system used by composers and performers when they compose, write, and play Western music. In a strict interpretation, all the scores generated in other cultural, historical, geographical contexts far from Western world contain non-conventional music notation: this would be the case of early medieval notation [1] – including *neumes*, still in use for Gregorian chant – or Indian *rāgas*, the melodic modes used in traditional South Asian music genres [2]. In the mentioned cases, scores seem non-conventional to Western non-experts only because they appear far from modern staff notation, but conversely there is broad agreement among practitioners on both notational rules and score-symbol performances.

Narrowing the field to Western culture and contemporary music, non-conventional notation appeared in order to address a range of experimental concerns. For example, during the first decades of the 20th century some innovations were introduced as a consequence of new composition and performance techniques, such as tone clusters and microtonal composing. In the 1950s and '60s – the golden age for graphic notation – the composers of the New York School (including John Cage, Morton Feldman, Earle Brown, and Christian Wolff) began experimenting with indeterminacy and investigated graphic notation as a way to restrict and reinvent the information provided to performers [3]. Back in 1969, John Cage and Alison Knowles collected an anthology of excerpts from hundreds of notated musical scores [4], often non-conventional as the one shown in Figure 1. Fourty years later, Theresa Sauer – a musicologist who turned to studying graphic design – acknowledged this work by publishing a new collection of notation examples close to the field of visual art [5].

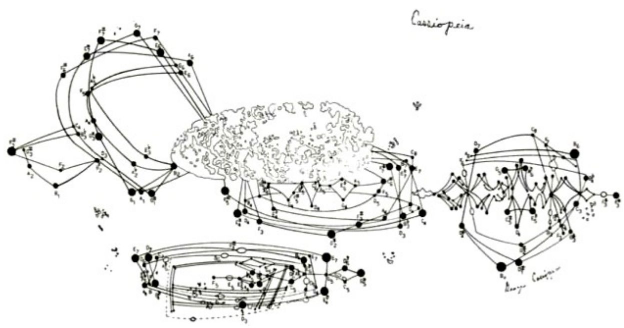


Figure 1: An example of non-conventional score: *Cassiopeia* by George Cacioppo.

The need for non-conventional notation was triggered not only by the artists' desire to conduct aesthetic experiments or break established rules, but also by a number of practical issues due to the advent of electronic and tape composition: new types of musical instruments – with unprecedented expressive potential concerning timbre, articulation, etc. – required brand new ways to encode scores. The issue of how to properly describe electroacoustic music has been addressed in many works, including [6, 7, 8].

The goal of this paper is to show how the power of Web interfaces can be leveraged to study, exploit and revive specific forms of non-conventional notation, providing support tools oriented to both music analysis and performance. Our case study is focused on *Black and White n.2*, a collection of 120 keyboard pieces written in 1969 by the Italian composer Franco Donatoni [9]. Noticeable example of aleatory composition, its notation recalls the principles of tablature, since the score indicates performance actions (specifically, instrument fingering) rather than pitches [10], thus leaving ample freedom to choose most music parameters.

This work is structured as follows: Section 2 addresses the problems related to the representation of non-conventional notation on a computer system and in a Web environment, Section 3 focuses on *Black and White n.2* and provides details to decode this aleatory-music score, and Section 4 discusses the Web interface implemented and publicly released as a research result.

This paper can be somehow referred to an earlier work entitled "Automatic Performance of Black and White n.2: The Influence of Emotions Over Aleatoric Music", which was presented during the 9th International Symposium of Computer Music Modeling and Retrieval held in London in 2009 [11]. That proposal concerned an automatic approach to extract emotion-related parameters from the analysis of a video, and described a computer system capable of mapping such results on the aleatory score of *Black and White n.2*, thus generating a suitable soundtrack. In this work the goals are different and varied, ranging from the rediscovery of Donatoni's aleatory music repertoire to the comprehension of his notation and the analysis of resulting computer-driven performances.

2. REPRESENTATION OF NON-CONVENTIONAL NOTATION IN COMPUTING

According to some experts, the printed score can be seen as a mediator of meaning. It is possible to identify two approaches to music notation typically followed by players: a *reproductive* approach and an *explorative* one [12]. In the former case, the function of the printed score is that of an explicitly normative document which prescribes how to play; in the latter, the function is an invitation to seek out implicit meaning according to the musicians' individual judgment. Both the approaches can occur only within a frame of agreed understanding shared with the composer, and this process can be challenging when notation rules are non-conventional, or their interpretation is intentionally left to the performer.

Due to the large variety of notations adopted in contemporary music, the representation of non-conventional scores – concerning both editing and computer-driven performance – is a huge and still open problem in the field of computing.

From the former point of view, namely editing, many music scores are far from CWN conventions and closer to figurative art, and score writers such as Finale, MuseScore or Sibelius – mainly conceived for common notation – are not adequate. In this case, different approaches are available in order to obtain a score in the digital domain: scanning a manuscript version, pushing the behavior of a traditional score editor as far as possible (e.g., by changing the default music font or inserting custom symbols as drawings), adopting a notation style that can be generated and edited through existing software (e.g., by using a graphics editor to combine the common notation produced by available score writers and graphic information entered via a digital drawing tablet), etc.

The other task, related to computer-driven performances of score symbols entered with one of the mentioned approaches, is hard to achieve as well. Narrowing the field to Web platforms, there are a number of experiments of simultaneous visualization and listening of non-conventional scores – for instance, there are dedicated YouTube channels – but often synchronization is hard-coded into the media and consequently these experiences do not offer any possibility of interaction. Conversely, institutions and research centers are more interested in investigating music processes, and the interfaces they design aim to highlight such processes in music notation and make them emerge through user interaction. From this point of view, an early experimentation via Web was conducted by INA-GRM: [1] interactive-listening examples are still available in the section named *Portraits polychromes*, unfortunately most of them require custom browser plugins such as Apple QuickTime and Adobe Shockwave.

A good compromise is offered by W3C-compliant [2] formats and Web platforms designed both to let untrained users enjoy music and to support music education and musicological investigation. Examples are music-oriented multilayer representation formats, capable to carry symbolic and structural information, multiple notation styles (including non-conventional scores), and audio information. For instance, the IEEE 1599 format is a standard conceived to describe and synchronize the information related to a music piece in all its aspects [13]. A Web player supporting IEEE 1599 documents is available at `http://emipiu.di.unimi.it`. This interface shows how non-conventional scores such as tablatures for lute, Labanotation for ballet, neumes for Gregorian plain chant, and even graphical scores for contemporary music can be linked to transcriptions in modern notation and synchronized with audio content, thus supporting interactive score following [14].

From this overview on available approaches and tools to write, edit, visualize and perform non-conventional notation, it is evident the need to design a context-tailored environment able to support a specific notation, thus releasing a solution customized for a given set of semantic and syntactic rules. This is the approach we will discuss in the next section to revive *Black and White n.2*, an aleatory music composition by Franco Donatoni.

3. AN EXAMPLE OF ALEATORY MUSIC: BLACK AND WHITE N.2

Black and White n.2 is a collection of 120 pieces written by the Italian composer Franco Donatoni. They can be played on any keyboard instrument, including piano, harpsichord, celesta, mute controller, etc. Performances for two and three keyboard instruments are allowed as well. This composition belongs to *aleatory music* – a definition first provided in [15] – since some primary parameters of the composition are not predetermined, rather their values depend on random processes or extemporary decisions made

[1] Institut National de l'Audiovisuel – Groupe de Recherche Musicales, `http://www.inagrm.com`.

[2] World Wide Web Consortium, `https://www.w3.org`.

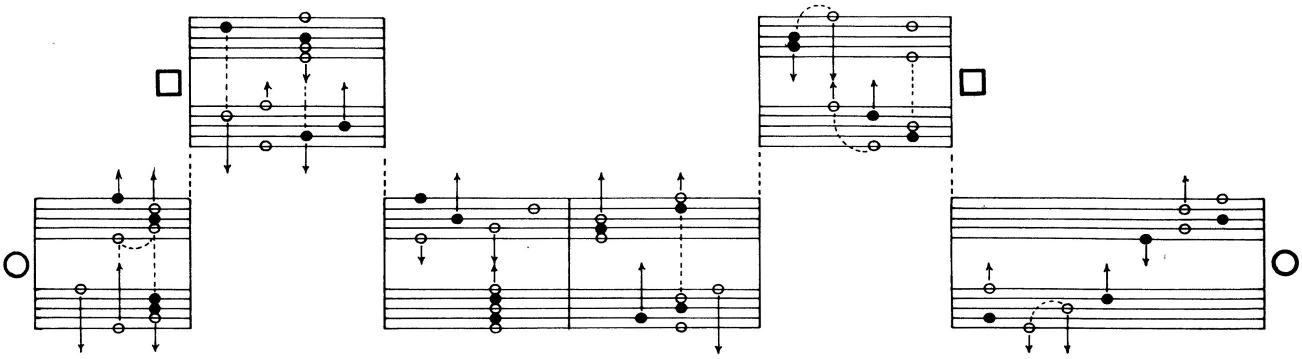


Figure 2: An exercise from the original score of *Black and White n.2*.

by the performer.

In the score preface, the author briefly explains the set of rules to read the score, significantly different from modern staff notation, as shown in Figure 2. First, the two staves usually assigned to standard keyboard notation (i.e. the grand staff) in this case do not carry pitch and rhythm information, but a sort of tablature for piano. Each staff line corresponds to a specific finger, consequently only staff lines (and not spaces) are allowed to host music symbols. For the right hand, the lower line corresponds to the thumb and the upper line to the little finger, and vice versa for the left hand. A note event can be represented as either a white or a black circle: the former indicates that a given finger should play a white key on the keyboard, the latter forces the corresponding finger to press a black key. Therefore, notation is mandatory in specifying which fingers should be used to press keys of a given color.

Conversely, the melodic, rhythmic and harmonic aspects of the composition – the information commonly carried by a traditional score – are left to the performer's extemporary interpretation. In order to understand the aesthetic and technical purposes of the composition, it is worth recalling the subtitle: "esercizi per le 10 dita", literally 10-finger exercises.

The instructions a player has to follow, expressed by the author himself in the work's preface, are:

- The association among symbol positions over lines and fingers is fixed;
- As it regards the color of note symbols, each circle can be either empty or filled, which forces the performer to play either a white or a black key respectively (using the indicated finger, in accordance with the previous rule);
- Each staff system is both preceded and followed by either an empty circle or an empty square. At the beginning of the performance the instrumentalist chooses the association of shapes to dynamics, i.e. if circles should correspond to ***ppp*** and squares to ***fff*** or vice versa;
- The concept of chord is associated to the vertical alignment of circles, possibly spanning over the two staves. In the latter case, chords will present the same dynamics and will be grouped by a vertical dashed line;
- Arrows pointing up or down can be specified chord by chord, thus providing a broad indication about the note range to use: an upward arrow for higher octaves, no arrow for the central region of the keyboard, a downward arrow for lower octaves.
- Dashed slurs provide suggestions about optional *legato* effects.
- Barlines have no specific meaning, since neither time signatures nor rhythmic values are explicitly indicated in the score. Nevertheless, barlines can be seen as a way to embrace a set of chords together, thus providing a sort of structural information.

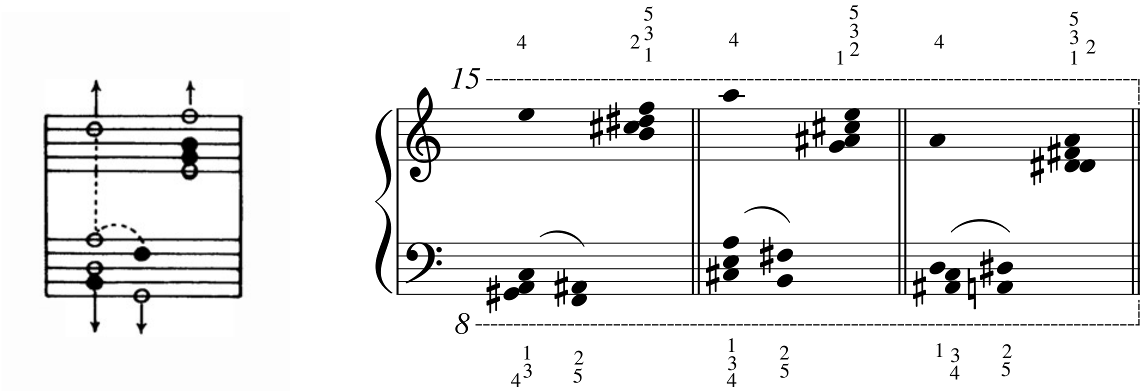
Since the mentioned set of rules leaves many music parameters to chance or improvisation, for a given score countless score instances and performances are possible. For example, Figure 3 shows a short excerpt from *Black and White n.2* and a number of notated instances that would all meet Donatoni's requirements.

4. ALEATORY MUSIC OVER THE WEB: BLACK AND BYTE N.2

In order to achieve the goal of reviving Donatoni's compositional intuitions and notation style, we have designed and implemented a Web interface called *Black and Byte n.2*.

The idea is to provide Web users with an intuitive interface specifically designed to explore notation, create new pieces compliant with Donatoni's rules, analyze possible score translations into modern notation, and finally play the resulting score. It is worth underlining that Donatoni rejected the idea of a prepared performance, and – in accordance with his ideas – our system has been conceived for analysis and training purposes, with no aesthetic or artistic goal. Nevertheless, our proposal addresses a wide audience, including keyboard performers, scholars in musicology, fans of contemporary music, and experts of music technologies.

In the following the design of *Black and Byte n.2* will be analyzed and discussed in all its aspects. As a result of our efforts, the project has been implemented and released at `http://blackandbyte.lim.di.unimi.it`.



(a) Excerpt (b) Some possible instances on a 4-octave keyboard

Figure 3: A short excerpt from *Black and White n.2* (a) and some playable variants in modern notation, with the mandatory fingering indicated above and below chords (b).

4.1 Principles of Design

The interface has been designed to take into account the different goals of this work:

1. Supporting Donatoni's notation, in order to rediscover a relevant piece of aleatory music;
2. Allowing the user to extend the original *corpus* of exercises. In our opinion this is perfectly compatible with the composer's original idea, since his creative and artistic work consisted in having established a set of conventions and rules to interpret his custom notation, and not in fixing specific instances of the score;
3. Helping the performer and the scholar in the understanding of the possible translations of the score in terms of modern notation. Since we are dealing with aleatory music, there are virtually endless combinations of pitches that are compatible with a given original fragment;
4. Providing a raw audio feedback to make the user familiarize with the aesthetics of this kind of music.

The layout shown in Figure 4 reflects the mentioned goals. The upper area of the interface contains some metadata (author, title, language) and a drop-down menu to open a multi-language description of rules, a short user guide, and some helpful editing tools. Below, the screen has been divided into two main parts: the upper one to enter Donatoni-style notation, as explained in the next subsection, and the lower one to show the corresponding modern staff notation. By clicking the finalize/reload button between the two sections, a new score instance in the lower area is produced. Finally, a simple media player is displayed at the bottom of the window.

4.2 Data Entry

All the score symbols supported by Donatoni's notation – well exemplified by the excerpt shown in Figure 2 – can be entered through mouse actions over a double grand staff.[3] Clickable positions are quantized with respect to a predefined grid, in accordance with the monospaced original notation.

Upper and lower grand staves contain the note events to be played ***ppp*** and ***fff***; the association of the former and the latter grand staff to music dynamics is left to chance or to performer's decisions. As for Donatoni's exercises, two-hand chords cannot be placed on different grand staves, i.e. only one dynamic level is allowed at any given time.

The main symbols to enter are the black or white circles representing note events, so the simplest mouse action has been assigned to this function: notes are placed through left mouse clicks on grand staves. Left clicks on already existing circles delete previous entries. Any click on a grand staff deletes the simultaneous note events possibly entered on the other. The mouse wheel lets user cycle between black and white note symbols.

Other supported symbols are: i) slurs, drawn by drag-and-drop actions starting on the first event and ending on the last event to tie, ii) up- or down-arrows, entered through left clicks above or below the chord to alter, and iii) additional barlines, placed by left clicks in the area between two consecutive grid positions.

The dashed vertical lines that connect simultaneous chords are automatically placed by the system as soon as two-hand chords are detected. Also the white circles and squares that embed the grand staves are rearranged on the fly on the basis of current entries.

4.3 Algorithmic Computation of Chords

This step can be seen as the automatic process that transforms the original score, entered in the first phase, into modern staff notation. Please note that score symbols are not merely converted from a kind of representation to another (like providing a modern transcription of neumes), but they must be inferred from the application of a set of generative rules. Consequently, the algorithmic computa-

[3] Please remember that here the grand staff is used to represent piano tablature instead of common notation.

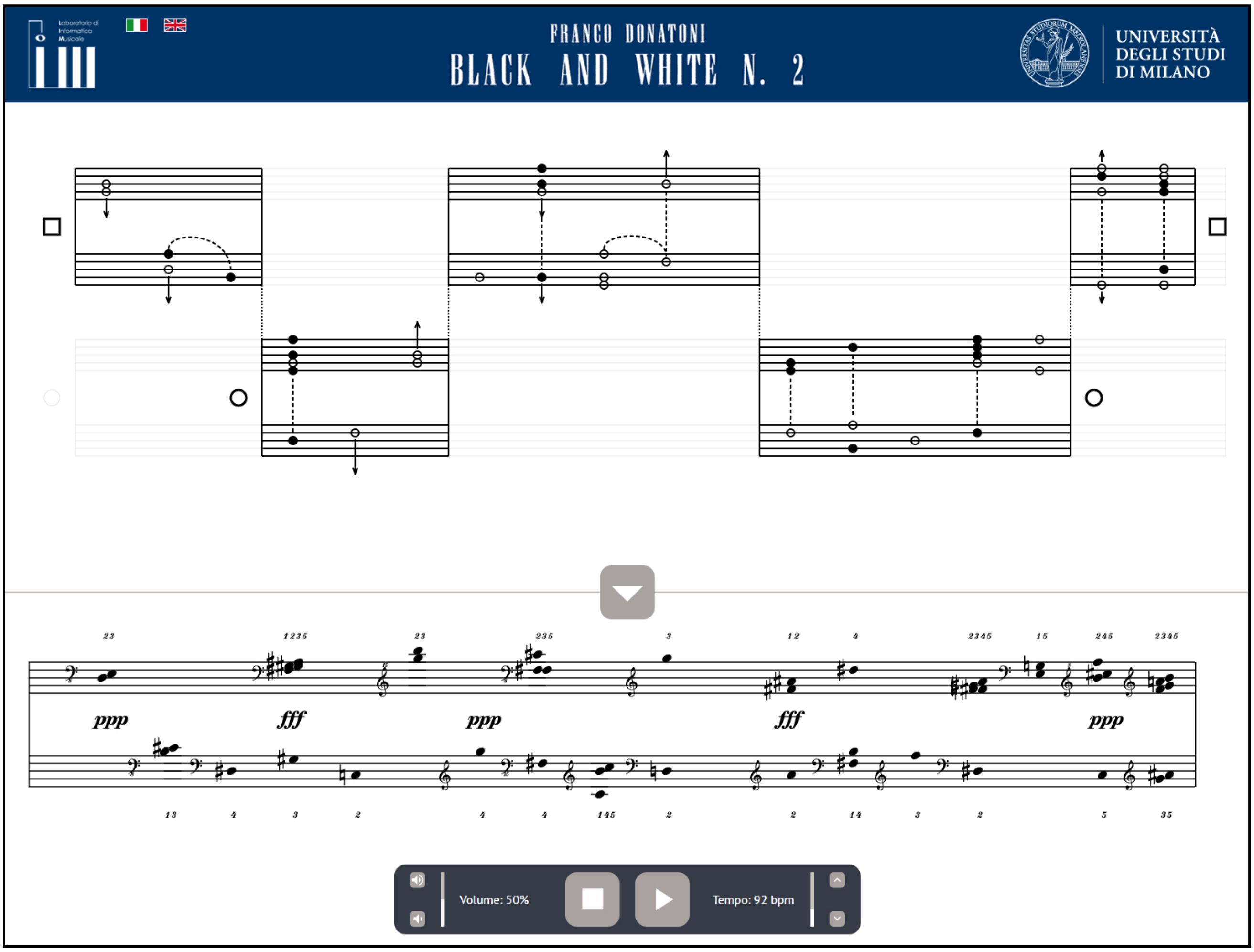


Figure 4: The web interface of *Black and Byte n.2*. The upper part shows a user-defined score encoded through the original notation by Donatoni; the lower part contains a random-generated instance of the score in common notation.

tion of chords requires to satisfy a number of non-trivial explicit and implicit constraints.

As it regards explicit constraints, the score clearly provides the performer with some mandatory indication, for example through instrumental fingering and arrows. Needless to say, not all chords formed by a compliant combination of white and black keys can be performed. Even if in *Black and Byte* the pianist is a computer system, virtually able to play any set of pitches, we decided to produce human-playable score instances. In order to reject unwanted chords, we modeled the hand positions that an averagely skilled pianist can take.

Another example of explicit constraint concerns the keyboard regions for chords, indicated in the score by up- and down-arrows. In this case, the requirements are easier to satisfy, thanks to suitable octave offsets for chords.

Other constraints are implicit and subtler: for example, when both hands have to play in the same region of the keyboard, the system has to detect possible overlaps and crossings and establish what a pianist can reasonably perform.

The process that brings to the choice of a sequence of chords is made of many steps. First, the system is provided with a set of chord models aimed at covering all the combinations of pitches that a pianist can easily perform. In order to have flexible but compact data structures, pitches in a chord are encoded in terms of semitone distances from the chord root, whose position is in turn movable in the keyboard. In other words, all the values that define each chord model are relative rather than absolute, and they can be instanced starting from any point of the keyboard. Inside data structures, chord models are clustered based on the number and type of fingers required for their performance.

Please note that the encoded models include not only "traditional" chords such as triads, sevenths and ninths in root position and in their inversions, but also more complex combinations of adjacent/non-adjacent piano keys, tone clusters, etc.

A preliminary selection of candidates is based on finger-related characteristics, that can be retrieved from Donatoni's notation. The following step consists in verifying if the selected chord model has at least one instance compliant with the white/black key configuration indicated in the score. If one of the mentioned steps fails, as there are no candidates having the required characteristics, a backtracking technique is used to select a new candidate. To adhere to Donatoni's concept of aleatory music, the choice of a specific chord model out of many compliant models, as well as the choice of the chord root among many compatible start pitches are left to chance.

From a graphical point of view, it is worth underlining that the production of a syntactically-correct and elegant notation is not easy to achieve. For example, the current clef has to change frequently in order to limit the number of ledger lines. Besides, the output score will typically contain a great number of accidentals, hard to be placed on staff due to note-head overlays and to be correctly shown/hidden in terms of printed and courtesy accidentals. Finally, an important part of the graphical representation is instrumental fingering, shown chord-by-chord either as a verification tool for the algorithm and as an aid to human performance.

4.4 Audio Performance

In order to produce a sound feedback, the Web interface has been equipped with a basic media player, capable of launching a computer-based performance of the score.

For our purposes, audio output is less important than the production of modern notation: in fact, the latter lets the user understand how the composer's ideas can be instanced on different sequences of logic events, an activity that has a high theoretical and musicological valence; conversely, the former is a mere translation of such computed symbols into audio events. Nevertheless, we decided to implement this function in order to give a broad idea of how the exercises from *Black and White n.2* or similar user-defined fragments could sound.

From a technical point of view, two Web-oriented approaches were possible: i) the adoption of the Web Audio API, and ii) the use of the Web MIDI API. At the moment of writing, both solutions are draft under development in the framework of W3C standardization activities.

The Web Audio API provides a powerful and versatile system for controlling audio on the Web, allowing developers to choose audio sources, add effects to audio, create audio visualizations, apply spatial effects (such as panning), etc. [16]. In order to obtain a high-quality output, this approach requires to load instrumental audio samples from the server. Sounds could be synthesized as well, e.g. through additive synthesis techniques, but the audio output would likely sound artificial.

The Web MIDI API specification defines an interface that supports the MIDI protocol, thus enabling Web applications to enumerate and select MIDI input and output devices on the client system and send and receive MIDI messages. The Web MIDI API is intended to enable MIDI applications by providing low-level access to the MIDI devices available on the users' systems [17]. Like in other MIDI-based applications, the resulting sound quality largely depends on the characteristics of the MIDI synth in use.

In addition to a different philosophy (i.e. server-side samples vs. client-side synthesis), an aspect to take into account concerns browser compatibility. As it regards the Web Audio API, most of its features are now available on all major browsers but Microsoft Internet Explorer, being supported e.g. by Google Chrome, Mozilla Firefox, Opera and Apple Safari in their desktop and mobile versions. Conversely, the Web MIDI API is currently supported only by Google Chrome, even if rumors say that also Mozilla and Microsoft are working on implementations and browser support should be guaranteed within the next year or two.

Moreover, the MIDI approach requires a virtual or physical MIDI chain, formed at least by a software synth installed on the client and equipped with a sound font.

Since cross-platform compatibility from our point of view is highly desirable, our choice fell on the Web Audio API. However, future developments in audio browser technologies could change such a design principle.

5. CONCLUSION

In this paper we addressed the problem of reviving Donatoni's *Black and White n.2* through an interactive Web platform, allowing musically untrained people as well as an expert audience to experiment with a relevant example of aleatory music.

Since at the moment of writing the Web prototype has just been released, an accurate experimentation phase has not been conducted yet, and some research questions are still open. For example: Is this approach effective to achieve the goals mentioned in Section 4.1? Was the interface properly designed to support human performance? Can additional tools be added to improve user experience, such as a step-by-step chord navigator? In the near future we will further investigate these aspects with the help of both untrained and expert users.

Even if we faced problems typical of a given compositional style (i.e. aleatory music) and specific for a given music piece (e.g., the set of rules to infer score instances, keyboard-notation issues, etc.), we think that our general approach and most technical solutions can be generalized to drive the design and implementation of similar platforms.

This activity was carried out in the context of a more general project aiming at the release of free Web tools to support music dissemination, education and analysis. Other applications are available in the Demo section of the LIM official Web site. [4]

Acknowledgments

The author gratefully wish to acknowledge their colleagues at LIM for their support. This project was made possible by the 2016-2018 research funding plan of the Department of Computer Science, University of Milan.

6. REFERENCES

[1] C. Floros and N. K. Moran, *Introduction to early medieval notation*. Harmonie Park Pr, 2005, no. 45.

[2] R. Widdess, *The rāgas of early Indian music: modes, melodies, and musical notations from the Gupta period to c. 1250*. Oxford University Press, USA, 1995.

[4] http://www.lim.di.unimi.it

[3] M. Nyman, *Experimental music: Cage and beyond*. Cambridge University Press, 1999, vol. 9.

[4] J. Cage, *Notations*. Something Else Press, 1969.

[5] T. Sauer, *Notations 21*. Mark Batty Pub, 2009.

[6] M. Helmuth, "Multidimensional representation of electroacoustic music*," *Journal of new music research*, vol. 25, no. 1, pp. 77–103, 1996.

[7] P. Couprie, "Graphical representation: an analytical and publication tool for electroacoustic music," *Organised Sound*, vol. 9, no. 01, pp. 109–113, 2004.

[8] K. Patton, "Morphological notation for interactive electroacoustic music," *Organised Sound*, vol. 12, no. 02, pp. 123–128, 2007.

[9] F. Donatoni, *Black and White N. 2 - Esercizi per le dieci dita per strumenti a tastiera*. Suvini Zerboni, 1968.

[10] M. Kennedy and J. Bourne, *The concise Oxford dictionary of music*. Oxford University Press, 2004.

[11] S. Baldan, A. Baratè, and L. A. Ludovico, "Automatic performance of Black and White n.2: The influence of emotions over aleatoric music," in *Proceedings of Computer Music Modeling and Retrieval : 9th International Symposium, CMMR 2012*, London, UK, 2012, pp. 437–448.

[12] C. Hultberg, "Approaches to music notation: The printed score as a mediator of meaning in western tonal tradition," *Music Education Research*, vol. 4, no. 2, pp. 185–197, 2002.

[13] D. L. Baggi and G. M. Haus, *Music navigation with symbols and layers: Toward content browsing with IEEE 1599 XML encoding*. John Wiley & Sons, 2013.

[14] A. Baratè and L. A. Ludovico, "An XML-based synchronization of audio and graphical representations of music scores," in *WIAMIS '07 Proceedings of the 8th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS '07 Proceedings of the Eight International Workshop on Image Analysis for Multimedia Interactive Services*. Santorini, Greece: IEEE Computer Society, 2007, pp. 64–67.

[15] W. Meyer-Eppler, "Statistic and psychologic problems of sound," *Die Reihe*, vol. 1, pp. 55–61, 1958.

[16] P. Adenot, C. Wilson, and C. Rogers, "Web Audio API," W3C, working draft, Dec. 2015, https://www.w3.org/TR/webaudio/.

[17] C. Wilson and J. Kalliokoski, "Web MIDI API," W3C, working draft, Mar. 2015, https://www.w3.org/TR/webmidi/.

THE 3-D SCORE

David Kim-Boyle
Sydney Conservatorium of Music
The University of Sydney
david.kim-boyle@sydney.edu.au

ABSTRACT

This paper examines attempts by composers to transcend the two-dimensional constraints of the printed page in musical notation. The author reviews how material depth in printed media has been explored to help create new structural forms and two of the author's works which feature real-time, three-dimensional scores are examined. Incumbent technical limitations and constraints of multidimensional notational schemas are discussed and the author concludes by arguing that the reading *through* of a notational schema affords a new spatial ontology for the works represented.

1. INTRODUCTION

In many respects, music notation can be regarded as a multi-dimensional construct that has evolved to facilitate communication of an increasingly complex and polyvalent musical language. The use of neumes, for example, while perfectly satisfactory for prescribing the pitch contours of plainchant does not suffice, nor is it intended, to convey complex rhythmic structure. Thus, the dimensionality of notational schemas is extended through the addition of new symbols such as time signatures and tuplets as musical language evolves.

Despite the growing complexity of musical notation, it has nevertheless always been bound by the constraints of the medium upon which it is inscribed. With the transition from parchment to paper in the mid-15th century, for example, the use of ancillary decoration becomes less pronounced.[1] The growing use of screen-based scores is also not without limitations with screen resolution, use of color, and speed of animation all representing constraints affecting communication with performers. [2] Whether paper or screen-based, both mediums have been jointly bound by the two-dimensional surface of the display. With the growing use of 3-D technologies in printing, imaging, immersive projection, and augmented reality systems it therefore seems natural to consider the creative potential of multi-dimensional notational schema. Such an appraisal, however, needs to be framed in the broader context of efforts to transcend the materiality of the printed surface.

[1] While there are obvious economic causes at play, the materiality of paper ultimately did not lend itself as well to the use of colored inks. [1]

2. TRANSCENDING THE PAGE

While the discovery of perspective in the 14th century allowed painters to more realistically depict depth on a two-dimensional surface, it was not until the mid-to-late 19th century with the development of photography and stereoscopic images that a more convincing illusion of depth was able to be conveyed to a viewer. Through the use of stereoscopic viewers to project phase shifted, and sometimes color corrected images to individual eyes, see Figure 1, early stereographic images created an overwhelming sense of presence for the viewer as was noted by early enthusiast Oliver Wendell Holmes in 1859 - "The first effect of looking at a good photograph through the stereoscope is a surprise such as no painting ever produced. The mind feels its way into the very depths of the picture. The scraggy branches of a tree in the foreground run out at us as if they would scratch our eyes out. The elbow of a figure stands forth as to make us almost uncomfortable." [3]

Figure 1. The Brewster stereoscope (left) and the Holmes stereoscope (right). Two early examples of stereoscopes from the late 19th century.

The desire to immerse the viewer in a scene, which played no small role in the early appeal of the medium, [4] has continued to be a driving force in the development of stereoscopic imagery today. This is most obviously notable in virtual reality or immersive systems such as the Oculus Rift [5] or, to a somewhat lesser extent, in the increasing popularity of 360 photos.

The illusion of depth that stereoscopic images create fundamentally represents an effort to embed more information about an image than can be ordinarily represented on a two-dimensional surface. Stereoscopic images, thus by their very nature, represent an effort to transcend the two-dimensional materiality of the page.

Many innovations in musical notation may similarly be framed in part as an effort to transcend the material constraints of the printed page. The use of colored ink in a number of scores of Baude Cordier and other composers from the Mannerist school of the 15th century to more clearly delineate rhythmic layers offers an early example, see Figure 2.

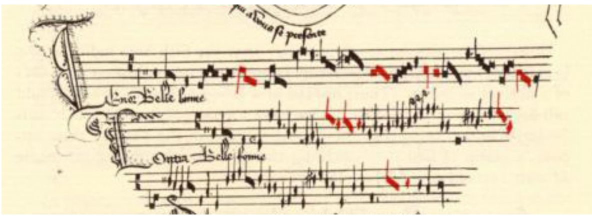



Figure 2. Detail from the score for Baude Cordier's (1380-1440) *Belle, Bonne, Sage* in which red colored notes denote rhythmic modification.

The use of screens to display musical scores also present many opportunities to transcend the constraints of paper-based scores with generative processes, animation, and extended graphical schema providing rich material for musical exploration. [2] To what extent, though, has material depth provided opportunities for creative exploration and the development of unique musical forms?

3. THE MATERIALITY OF DEPTH

Unlike visual art-forms in which the illusion of depth is typically motivated by a desire to immerse a spectator in a scene, notational depth in music notations and representations is most commonly driven by a desire to embed additional layers of information in a notational schema. There are, however, deeper ontological motivations behind these explorations which will be touched upon later in this paper.

Perhaps the most well known early example in which the materiality of depth plays an integral role in defining musical structures are those works of John Cage which involved the use of transparencies, the most famous of which include the *Variations II* (1961), *III* (1962), *IV* (1963), and *VI* (1966); *Fontana Mix* (1958), *Music Walk* (1958) and *Cartridge Music* (1960). In each of these works musical structures emerge from the superimposition of preprinted transparencies and printed sheets, see Figure 3.

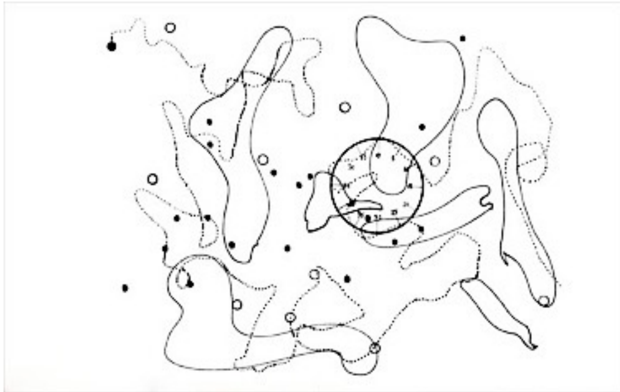


Figure 3. A performance score for Cage's *Cartridge Music* (1960). The score is created by superimposing four transparencies on one of up to twenty pre-printed sheets.

Musical form in these works is an emergent result of an aleatoric process, a sonic assemblage or composite of discrete individual prescriptive actions. [6, 7] While Cage's interest in the use of transparent media for musical composition appears to have come to a close in the 1960s, the potentiality of depth for creating unique forms continued to be explored in his lesser known visual art works built from plexiglass such as, for example, *Not Wanting to Say Anything About Marcel* (1969), see Figure 4, in which randomly chosen words and letters are distribued across parallel sheets of plexiglass according to the results of a three-coin toss. [8]

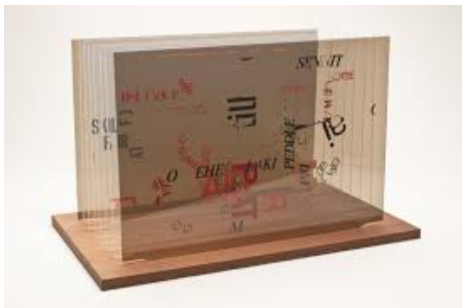


Figure 4. *Not Wanting to Say Anything About Marcel* (1969).

The influence of Cage's use of superimposed transparencies was felt more strongly in Japan where, as is well known Cage spent considerable time in the 1960s, more so than in Europe or North America. In Toshi Ichiyanagi's *Music for Piano No. 7* (1961), for example, the performer is provided with ten pre-printed sheets with three options for arranging them to form a performance score. The third option requires the performer to – "3. Accumulate ten sheets freely in a row. So each sheet is read only in part. The performer may change the order or turn the score upside down in the same position to continue the piece." [9]

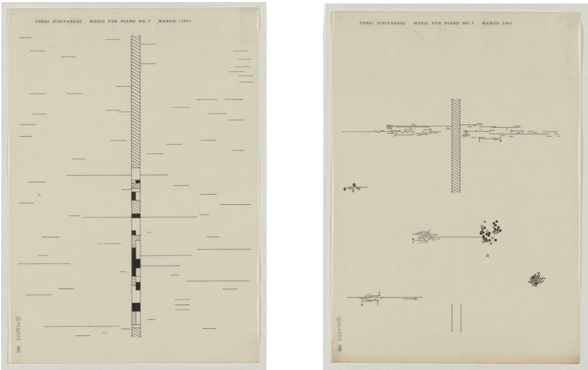


Figure 5. Two pages from preparatory material of *Music for Piano No. 7* (1961) by Toshio Ichiyanagi.

Each of the ten sheets available for the performer employs graphic notation often featuring a long thin rectangular prism which bisects the page and is filled and surrounded by various shadings, lines, circles and squares which in turn designate pitches, pitch ranges and various harmonics to be performed. The random accumulation of sheets naturally creates potential musical structures which unfold as the pianist reads *through* the sheets.

The creation of musical structure through material depth also plays a fundamental role in Toru Takemitsu's *Corona* (1962). The score for this work, designed in collaboration with graphic designer Kôhei Sugiura, [10] exists in two versions, one for solo piano and the other for string orchestra. It consists of five cards and transparencies printed in different colors which may be interlocked in configurations of the performer's choosing to form a performance score, [10, 11] see Figure 6. Both versions of the work present performers with a great deal of freedom although Takemitsu does provide detailed instructions on how the graphic shapes and symbols are to be interpreted.

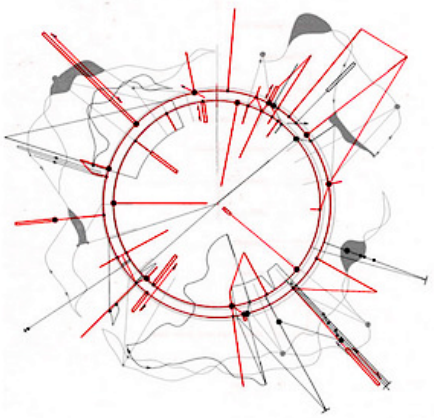


Figure 6. A performance score for the Takemitsu's *Corona* (1962) for solo piano.

Like arrangement number three of Ichiyanagi's *Music for Piano No. 7*, and Cage's *Cartridge Music* the performance score for *Corona* constrains prescriptive information to a single "page". This is also a notable feature of many works featuring scores created in real-time as discussed in [12] where the concept of a page turn is an obvious anachronism.

Notational depth is also a feature of Kenneth Gaburo's *Lingua II: Maledetto* (1967-8) and Herbert Brün's *Mutatis Mutandi* (1976) the scores for each of which are constructed from complex superimpositions of words, letters, and other graphical shapes, see Figure 7.

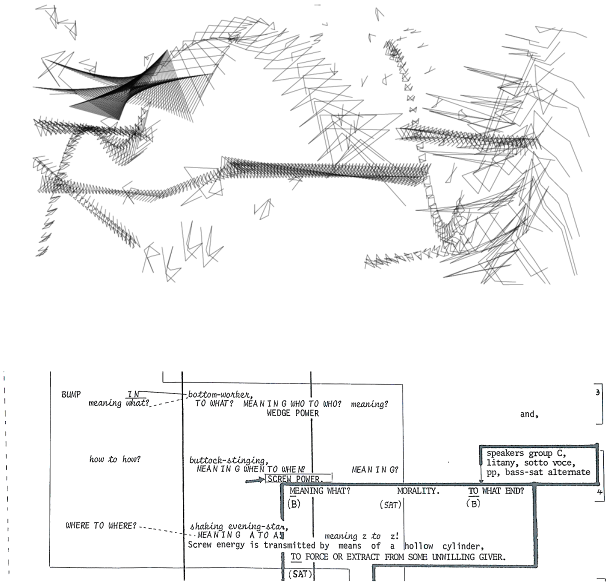


Figure 7. Score excerpts from Herbert Brün's *Mutatis Mutandis* (top), and Kenneth Gaburo's *Lingua II* (bottom) both of which employ complex graphic superimpositions to create musical structure.

In each of these works the performer is not afforded the flexibility to create their own multi-layered score from the superimposition of pre-composed pages or transparencies. Nevertheless, the printed superimposition of graphical and typographical shapes presents similar authorial intentions and interpretive challenges. Of these, Brün has written – "The interpreter, now, is to construct, by thought and imagination, HIS version of a structure that might leave the traces which the graphic displays. The interpreter is not asked to reconstruct my computer program, the structured process that actually generated the graphics. Rather he is asked to construct the structured process by which HE would like to have generated the graphics." [13] The notational complexity of Brün's work with its overt layering of graphical planes, naturally encourages performers to read *through* the score in addition to a more accustomed reading *across*. The multi-dimensionality of the notational structure is also foregrounded in Gaburo's work for seven virtuoso speakers with simultaneity and overlapping of vocal enunciations mirrored in notational depth and textual superimposition.

Multi-dimensional notational structures, especially those exploring depth, naturally push the boundaries of what can be represented on a two-dimensional surface. It is perhaps not surprising then that composers such as Cage and Gaburo moved on to explore multidimensional multimedia forms in works such as Cage's *Musicircus* (1967), *Roaratorio* (1979) or Gaburo's *My, My, My, What a Wonderful Fall!* (1974) after having largely exhausted the musical possibilities of notational depth on printed media, [2] while contemporary artists such as Marc Berghaus [15] and Martin Daske [16] have developed three-dimensional sculptural scores as a means of exploring the use of depth in creating unique musical forms. Before closing this section, it is also worth acknowledging a body of concrete poetry by poets such as Jackson MacLow, bp Nichol, and Augusto de Campos whose work was fundamentally invested in exploring the relationships between typographical structure and literary expression. [17]

4. 3-D SCORES

Not surprisingly, with the rapid evolution of computer processing power and the parallel development of programming languages for real-time graphic display and processing such as Jitter, Processing, and OpenFrameworks, the use of transparencies, slides and other media suited to representing multiple layers of notation has ostensibly ceased. In the author's own creative work, Jitter has been a core tool that has afforded exploration of the potential of 3-D notations. As of writing, two of the author's works, with another in development, have featured 3-D scores. In each of these works, the score is generated live and employs a unique graphical schema.

4.1 point studies no. 2 (2013)

point studies no. 2, for any two pitched instruments, presents performers with a series of colored nodes randomly distributed across a three-dimensional grid. [3] The color and relative size of a node denotes the pitch and intensity respectively of a note to be performed. Performers are free to determine the sequence of pitches performed by choosing various pathways through the nodes, which are connected by thin lines according to their alignment, see Figure 8. A "multi-player" computer-generated interpretation of the score, consisting of sine tones, accompanies the performer's interpretation.

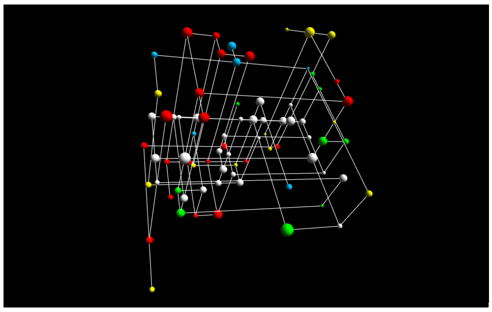


Figure 8. Screen capture of the score for *point studies no. 2* (2013) for any two pitched instruments and computer.

The score for *point studies no. 2* is generated live at the start of each performance but once generated, the pitches, their intensities, and the possible navigational pathways between them remains fixed. No position within the grid of nodes is privileged, however, as the performers are free to choose their starting node and respective pathways through the score. In order to facilitate legibility of the full distribution of pitches, a virtual camera moves around, in-and-out of the score.[4] The illusion of depth within the score is created by distributing nodes in three dimensions where mapping in the z-dimension is achieved by adjustments of scale and lighting. Like any three-dimensional object presented on a two-dimensional surface, occlusion of background layers presents a challenge difficult to overcome. To that end, the author is exploring how augmented reality systems might offer opportunities for performers to physically engage with a score by integrating choreographed movement around the virtual score within the interpretation.

4.2 16:16 (2016)

16:16 (2016) for prepared piano four hands, presents a different approach to the use of three-dimensional notation and is the first work of the author's to explore the use of stereoscopic imagery. While, the work has some superficial similarities to *point studies no. 2* in that colored nodes are used to represent pitched events, *16:16* extends the graphical schema considerably and presents the performers with a more complex range of possibilities in interpretation.

In *16:16*, nodes represent various pitched events with color denoting a different type of preparation material (red⟶screw, green⟶rubber, blue⟶plastic, yellow⟶metal, white⟶wood). To facilitate interpretation, it has proven helpful for small colored adhesive labels to be applied to the piano keys, see Figure 9a. Nodes are randomly distributed on a rectangular grid which contains various numbers within certain squares indicating the number of times a note is to be repeated, see Figure 9b. During the work, nodes rise and fall from the fixed grid

[2] Curiously, Jim O'Rourke's 2006 recording of Takemitsu's *Corona*, also foregrounds multidimensionality through juxtaposing two different performances of the work. [14]

[3] A score excerpt is available for viewing at <http://www.davidkimboyle.net/point-studies-no-2-2013.html>.

[4] This virtual camera corresponds to the OpenGL camera which looks upon a scene to be rendered within the Jitter environment in which the work is programmed.

with one of the pianists interpreting those that rise and the other those that fall. Such nodes are said to be activated and are the only nodes eligible for performance by either player.

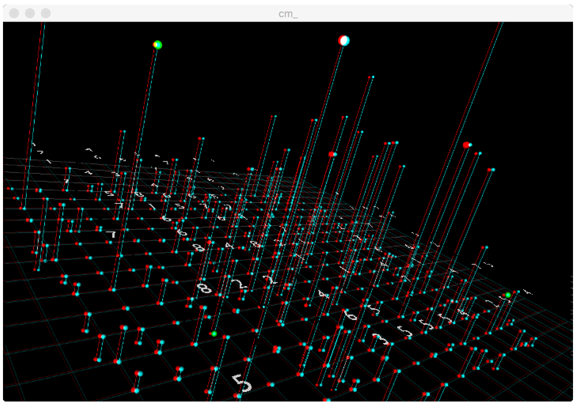


Figure 9. a) Colored adhesives applied to piano keys to facilitate score interpretation (top), b) Screen capture from the anaglyphic score for *16:16* (2016) for prepared piano four-hands (bottom).

Like *point studies no. 2*, the virtual camera through which the displayed score for *16:16* is presented to the performers moves in three-dimensional space which not only facilitates legibility but also helps reveal and occlude particular segments of the score. The real innovation of *16:16*, however, is that it is presented as a stereoscopic, anaglyph image requiring each of the performers to wear red-cyan glasses. In an anaglyph, a stereoscopic image is created by physically separating the red and cyan channels of a full-color image. In order to ensure that each eye only receives the filtered image intended for it, red-cyan glasses need to be worn. The further the channels are separated, the greater the illusion of depth that is created. In the Jitter/OpenGL environment within which the score for *16:16* is generated, this is achieved by rendering a scene from two different camera angles each of which is subsequently filtered to display discrete red (left) and cyan (right) channels, see Figure 10.

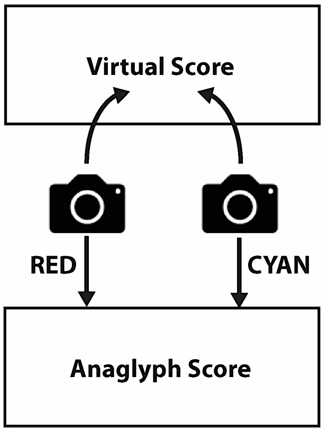


Figure 10. Creating a stereoscopic image in *Jitter*.

Unlike stereoscopic images which require the use of polarized light, or immersive systems which radically constrain interaction with an instrument, the anaglyph image offers a cheap method through which a stereoscopic effect can be created for performers. In *16:16*, it facilitates legibility of the three-dimensional movements of nodes and helps distinguish nodes aligned along nearby axes. The use of an anaglyph image is not, however, without constraints. It is somewhat pointless, for example, to display such imagery to an audience as is often done in concerts featuring works with real-time screen scores unless members of the audience are presented with their own red-cyan glasses. Of more concern, however, is the fact that anaglyph images have a limited color field with which they are effective. [18] While this has not been a tremendous constraint in *16:16* with a color field limited to red-green-blue-yellow-white-black, it still represents a constraint affecting typographical choices.

5. CONCLUDING THOUGHTS AND FUTURE WORK

By promoting a reading *through* rather than just a reading *across* within a notational schema, scores that feature juxtaposition, superimposition, and other three-dimensional techniques help establish a new spatial ontology for the works they afford. [19] Through offering new ways of engaging with space, these works also draw awareness to the activity that takes place within it. Cage strongly hints at this in his discussion with Richard Kostelanetz on *Variations III* where he states – "…We are constantly active; we are never inactive. There is no space in our lives. But there is a greater or lesser number of things going on at the same moment; so that if I'm not doing anything other than listening, the fact that I'm listening is that I'm doing something by listening. That's what *Variations III* is." [20]

This reinvigorated awareness of the social dimension of space was also strongly present in much of the 1950s Japanese avant-garde tradition as represented in the work of Ichiyanagi, Tone and others who explored multidimensional notational schemas. [21] And in some respects, it also hearkens back to what Eco was referring to in his discussion of Baroque poetics where he argues that "…the Baroque work of art…induces the spectator to shift his position continuously in order to see the work in constantly new aspects, as if it were in a state of perpetual transformation." [22]

While many pragmatic issues still need to be addressed, perhaps the area with the most potential for future development in three-dimensional representations are those afforded by immersive and augmented reality systems. Concurrent with that opportunity however, is the challenge of overcoming the isolationist tendencies inherent in most such systems. Nevertheless, the author believes the new spatial ontologies afforded by such systems is rich in aesthetic potential with exciting opportunities for new forms of musical engagement.

6. REFERENCES

[1] R. Rastall, *The Notation of Western Music*. St. Martin's Press: New York, 1982.

[2] C. Hope & L. Vickery, "Screen Scores: New Media Music Manuscripts," in *Proceedings of the International Computer Music Conference 2011*, Huddersfield, pp. 224-230, 2011.

[3] O. W. Holmes, "The Stereoscope and the Stereograph," *The Atlantic*, June 1859. Accessed September, 2016. Available at <http://www.theatlantic.com/magazine/archive/1859/06/the-stereoscope-and-the-stereograph/303361/> .

[4] L. Spiro, "History Through the Stereoscope: Stereoscopy and Virtual Travel," October 30, 2006. Accessed September, 2016. Available at <http://cnx.org/content/col10371/1.3>.

[5] <http://www.oculus.com/> 2016. Accessed September, 2016.

[6] D. Kim-Boyle, "The Ontology of Live Notations through Assemblage Theory," in *Proceedings of the TENOR2016 Second International Conference on Technologies for Music Notation and Representation*, Cambridge, 2016. Available at <http://tenor-conference.org/proceedings/2016/13_Kim-Boyle_tenor2016.pdf>.

[7] Y. Manolopouluo, *Architectures of Chance*. Routledge: London, 2013.

[8] M. Perloff & C. Junkerman (Eds.), *John Cage: Composed in America*. University of Chicago Press: Chicago, 1994.

[9] T. Ichiyanagi, *Music for Piano No. 7*. Edition Peters: London, 1961.

[10] P. Burt, *The Music of Toru Takemitsu*. Cambridge University Press: Cambridge, 2006.

[11] C. Deupree, "Sound and Vision: Takemitsu's Corona," October 7, 2009. Accessed October, 2016. Available at <http://neojaponisme.com/2009/10/07/sound-and-vision-takemitsus-corona/>.

[12] D. Kim-Boyle, "Visual Design of Real-Time Scores," *Organised Sound*, Vol. 19, No. 3, pp. 286-294, 2014.

[13] H. Brün, *Mutatis Mutandis*. Smith Publications: Baltimore, 1968.

[14] J. O'Rourke, *Corona – Tokyo Realization*. Columbia, COCB-53573, 2006.

[15] M. Berghaus, "Some Ideas for Three-Dimensional Musical Scores," *Leonardo Music Journal*, Vol. 21, pp. 7-8, 2011.

[16] M. Daske, <http://www.tribordstudio.de> Accessed February, 2017.

[17] E. Williams (Ed.), *An Anthology of Concrete Poetry*. Something Else Press: New York, 1967.

[18] I. Ideses & L. Yaroslavsky, "New Methods to Produce High Quality Color Anaglyphs for 3-D Visualization," in *International Conference ICIAR 2004*, Porto: Portugal, pp. 273-280, 2004.

[19] M. Fowler, "Reading John Cage's *Variations III* as a Process for Generating Proto-Architectural Form," *LEONARDO*, Vol. 45, No. 1, pp. 34-41, 2012.

[20] R. Kostelanetz, *Conversing With John Cage*. 2nd Edition, Routledge: New York, 2003.

[21] C. Merewether & R. I. Hero (Eds.), *Art, Anti-Art, Non-Art: Experimentation in the Public Sphere in Postwar Japan, 1950-1970*. Getty Publications: Los Angeles, 2007.

[22] U. Eco, *The Open Work*. A. Cancogni (Trans.). Harvard University Press: Cambridge, 1989.

AN ARCHITECTURAL APPROACH TO 3D SPATIAL DRUM NOTATION

Jeremy J Ham
RMIT University
jeremy@surfcoastarchitecture.com.au

ABSTRACT

This research has evolved from creative practice focused on inter-disciplinary positioning between the domains of music and architecture. Through engagement in the theories and practice of architectural representation and the computational tools of spatial design, a new form of 3D spatial drum notation is presented. This notation seeks to compliment the capacities of traditional drum notation and overcome issues inherent in a theoretical 'musico-perspectival hinge' between the notation and the meaning of the notation. A representational schema of the spatial drum notation is discussed in the first instance in relation to the development of a lexicon of referent drum patterns and phrases and then in the testing of notation on a multi-layered improvised 'drumscape' composition. The paper culminates in the extension beyond notation into the realm of music spatialization through 3D printing, digital fabrication and Virtual Reality.

1. REPRESENTATIONS OF MUSIC AND ARCHITECTURE

The proposition that the field of architectural representation can inform domain of musical notation draws on creative practice PhD project work in music and architecture. The research draws upon the author's 30 years experience as an improvising drummer, as an architect with around 30 years experience in designing buildings and University educator researching design and representational media. By bringing together these practices in the form of a post-Xenakian integrated 'musico-spatial design practitioner', here is much fertile ground for exploration in both domains, and the space in-between.

This paper represents a first foray into the field of musical notation from a base expertise in architecture. From this position, a founding question arises for this paper directed at a musical notation conference: 'How can the theory and practice of architecture provide new insights into the field of musical notation?' In order to answer this, one must establish an outline of what architects do, and how they represent their creative practice.

In essence, the job of an architect is to transform a functional design brief relating to a site and the needs of people, generate spatial ideas, represent these ideas in the form of a resolved building design and communicate them to people for review; document the design using representational media then facilitate the physical construction of the design into built form on the site.

Central to the process is the use of representational media to form representations of ideas under development to 'achieve situational awareness that allows for meaningful criticism of design [1]'. Representational media constitute *analogue* or *physical* systems (tracing paper, graphite and ink) or *digital* or *virtual* systems (involving scanning, Two Dimensional Computer Aided Design (2D CAD), Three dimensional Computer Aided Design (3D CAD) modelling, animations and rendering). These are used during various stages of the design process to inform design and to communicate ideas separately or in hybrid combinations [2].

Many architectural practices utilise ArchiCAD™ or similar programmes to design buildings through modeling and drawing. Objects (walls, roofs etc.) are generated in plan (Figure 1, top left), edited in 3D (top right), then worked up with notes, lines and fills to form a set of 2D drawings (bottom left and right) for emailing or printing. Drawings and models can be zoomed, rotated, sliced in multiple ways in order to enable comprehensive understandings of the design. The modality of operation, where a spatial object is design in three-dimensions from different planes, is entirely natural for most spatial designers experienced in the use of 3D CAD software such as Revit™, Rhino3D™ and 3D Studio Max™. It is this defining characteristic that forms the basis of this research.

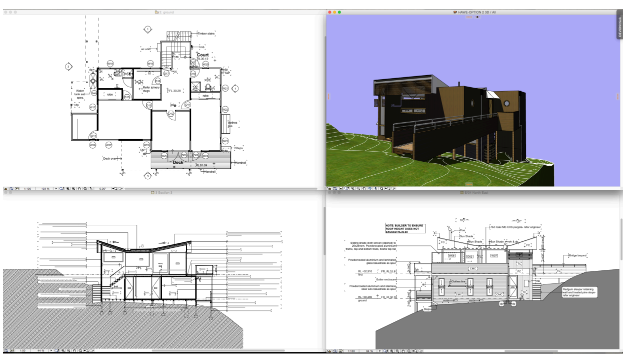


Figure 1. ArchiCAD screen print of a building design showing plan (top left), 3D perspective (top right), sectional drawing (bottom left) and elevation (bottom right).

Pérez-Gómez and Pelletier [3] developed the concept of the ‘perspectival hinge’ relating to how two-dimensional representations in plan, elevation and section form a hinge for understanding (or lack thereof) of the three-dimensional objects they represent. This ‘invisible perspectival hinge is always at work between these common forms of representation and the world to which they refer’, thus acting to limit comprehension in design processes. *Ideas* of buildings are built up between a set of projections (plan, section, elevation, perspective). This idea of the building is then translated into a building, usually by a third party (builders). Thus, it follows that the ideas under development may be limited by the two dimensional nature of the medium of 2D drawing. Working beyond the limitations of the perspectival hinge requires training and experience and is particularly relevant for students of architecture as novice designers [4].

Many lay people cannot read architectural drawings, just as many non-musicians cannot read musical notation. Whereas architects spend years learning the art of representation, the ability to read music is not intrinsic to the playing of music. Novice architectural students often struggle to understand the basics of their own designs, and must work their way through the limitations of this perspectival hinge when working in 2D. Mature practitioners of architecture expertly translate two-dimensional representations into perceptions of the three-dimensional object being represented. Many expert musician practitioners such as Jimi Hendrix, BB King have however navigated their musical world outside of a notation system and perform by memory and ear.

It seems plausible that the two-dimensional and symbolic nature of traditional musical notation acts as a hinge to the understanding of the music that it represents. This ‘musico-perspectival hinge’ is where *ideas* of music are built up through the placement of notes on a stave, which is then translated into a musical performance by trained musicians. Those with sufficient training are able to expertly translate the symbolic conventions into musical events in time, performed on an instrument. For the untrained or those with limited training, the symbolic schema of traditional notation is either meaningless or require significant time and effort to interpret.

The speculative question arises as to the relationship between the spatiality of the instrument (i.e. the issues inherent in the spatial engagement of the musician in making music on the three dimensional musical instrument) and the spatiality of the notation as a means of providing instructions for musicians or in informing the analysis of completed musical works. The playing of a piano, for example, is confined within limited spatial boundaries. The spatiality of the keyboard is intrinsically linear, thus more directly translates into the linearity of traditional musical notation than, say the drum kit.

The spatiality of the drum kit, as a set of 3D instruments (drums and cymbals) positioned in physical space played by the musician with two hands and two feet using drum sticks, is a core consideration in the spatial drum notation described below. For example, the digital drum kit that forms the basis of this research comprises six drum pads, four cymbal pads, a double kick pedal operating a bass drum and a pedal-operated hi-hat positioned within a spatial envelope of approximately 2.0 metres wide x 1.5 metres deep x 1.5 metres high. The kit is played from a pivot point of a drum stool with the snare drum, hi-hat and bass (kick) drum forming the core, with other drums and cymbals played in linear and radial patterns around the instrument (see figure 2).

The question arises as to the appropriateness of traditional notation to represent drumming. As Stone [5] states: ‘Musical notation, after all, is not an ideal method of communication, utilizing, as it does, visual devices to express aural concepts. But it is all we have’. The mapping of these instrumental engagements into the linear and two-dimensional spatiality of traditional notation would seem to further any theoretical ‘musico-perspectival hinge’ that exists between the notation and the instrument for which the notation exists. Whilst this may not be an issue for simple drum music, the issues with traditional notation for the drum kit are even more pronounced when dealing with complex, polyrhythmic drum patterns or even non-quantized basic 4/4 beat with the subtle slurs that constitute individual style (see Figure 3). Quantization of the MIDI file makes the traditional drum score more readable, however removes information related to the individual stylistic elements of play, for example playing behind and ahead of the meter and the dynamics of drumming.

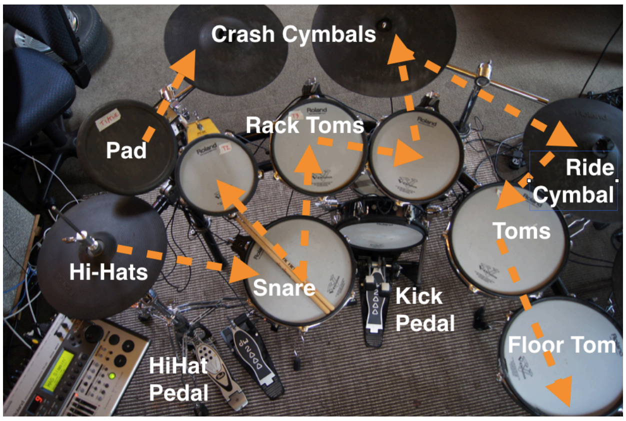


Figure 2. Digital drum kit arrangement, indicating some ideas of directionality and spatiality of play.

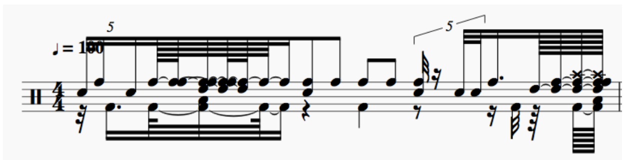



Figure 3. MuseScore score of basic 4/4 drum pattern with accents and slurs that constitute individual playing style.

Pérez-Gómez and Pelletier’s perspectival hinge appears to be related to the theory of ‘affordance’ put forward by Gibson [6] that describes the interactions between people, objects and actions. Norman [7] applies this concept to design, with the principal being that affordances should provide users with strong clues to the function of things. Whereas ‘the purpose of using a musical notation may be obvious, the notation’s meaning itself is not always so apparent [8]'. From this foundation, an implementation of a spatial notation for the digital drum kit is outlined that offers an architectural approach to drum notation that may provide affordance to the structure of improvised polyrhythmic drum music.

2. SPATIALIZING NOTATION AND THE "Y-CONDITION"

Performed in real time, music never exists as a whole at any given moment, but rather unfolds in a linear manner over time, and assumes an entity only in retrospect, in the memory of the listener or the performer. However reading a compositional music score is a process closer to perceiving space, as it exists as a whole at any given moment but may be retained by the observer only by a process of observation over time, walking around through, and above it. [9].

Architects invent notation systems to support design processes, to communicate these to others and to make artful representations of speculative ideas [10]. Daniel Leibskind's 'indeterminate spatial diagrams' of his chamber works act as speculative representations 'not regulated to site, scale, orientation, ground and other usual architectural references [11] and as such they constitute diagrammatic forms of art that describe spatiality. Similarly, Bernard Tschumi's 'event notation' and Parc de la Villette fireworks notation 'approach practices that are characteristically non-notated (at least in a temporally precise manner) with a view to codifying and communicating a particular instance of those practices [12]'.

Architect-engineer and composer, Iannis Xenakis provides many examples of where unique notation was invented to provide meaningful representations of design ideas. Xenakis set the standard for musico-spatial design creative practice modalities and the invention of notation systems to support his electroacoustic compositions, polytopes and other creative endeavours. His notation in Pithoprakta, Metastasis and other examples reflected his training as an engineer and mathematician- with lines, vectors, points and other graphical elements directing performers, instruments, sounds and actions. John Cage similarly dismissed conventions of temporal structure, repetition and proportional counting to create indeterminate soundscapes. His notation reflects an approach of 'blurring of unprescribed music, environmental sounds and dismissal of established rules in notation and performance [11]'. Cage's fascination for graphical, non-standard notation systems and random elements is evidenced in his book, with the book itself generated by chance operations [13].

Many notations operate as two-dimensional representations of multi-parameter musical compositions comprising multiple instruments performing complex operations in time and space. Many composers have sought the elusive third dimension in musical notation. Rebelo [12] describes 3D scores as 3D objects to be viewed and interpreted by performers from different directions. This third dimension is still mediated by the 'hinge' of the computer screen interface, however the digital format offers a range of possibilities to develop graphic notation practice by incorporating colour, real time generation, video and interactivity. A summary of approaches of visual notation in the 'visual/sonic representation continuum' is offered by Hope, Vickery [14]'. Vickery's 'rhizomatic' score for 'Sacrificial Zones' engages in the third dimension through a series of layered planes of visual representations of sound. This engagement in the third dimension in musical notation is being explored by many music, performance arts and other creative practitioners as they seek new ways to engage in creative media. It is proposed that architects may have something to offer the music community of practice in this area.

Architects, who have long been fascinated with inter-disciplinary connections with music [15], have sought ways of engaging in the 'architecture as frozen music' paradigm. Elizabeth Martin describes the "Y-Condition" as the theoretical intersection between domains where 'there exists a definable membrane through which meaning can move when translating from one to another [16]' The computer has enabled this cross-fertilization whereby the 'reduction of all information to a binary signal, be it a picture, a text, a space or a sound - all data is recorded as a binary sequence allowing computation as defined by programming languages and communication through networks according to transmission protocols' [17]. The principal that 'the byte shall be the sole building material [18]' acts to enable compositional, and therefore notational, opportunities within the spatial dimension.

Mediating this 'y-condition' in between music and architecture computationally requires the 'practiced hand' of the digital craftsperson [19]. Computational processes have been adopted by Ferschin, Lehner [20] the 'Spatial Polyphony' analysis of Bach's fugues by Christensen [21], the shape analysis of Krawczyk [22], the many speculative theoretical, philosophical and computational investigations into 'liquid architecture' of Marcos Novak [23] and the wide range of 'musical sculptures' and music-architecture explorations of Jan Henrik Hansen [24].

My interest lies not in creating a 'frozen music' but enabling ways of 'freezing' the process of music creation in the spatial dimension to create a spatialized notation system to provide meaning to elements of my creative drumming practice.

3. SPATIAL DRUM NOTATION

Rebelo [12] defines the roles and function of notation for performance, composition, design, choreography, gastronomy and architecture as being for the purposes of documentation, communication, for reflection and in the production of new works. I am interested in spatial notation as a means of exploring this "Y-Condition" and to provide an inter-disciplinary perspective on music and music notation to engage in reflection-on-action [25] and develop understandings of my creative practice as a drummer.

As with Tschumi, Xenakis and many others, I have invented a notation schema to serve a creative practice agenda. In my case this spatial notation system is for the digital drum kit using the Computer Aided Design tools I use in my architectural practice. The spatial notation is primarily directed at my own practice, however in pursuing my own personal notation system it is hoped that insights can be provided for others- with or without similar training in spatial design.

The spatial drum notation is derived from a definition scripted in Grasshopper™ for Rhino3D™, a parametric CAD tool commonly used in complex and innovative spatial design work. The 'ImprovSpace' definition reads time, note, note duration and velocity MIDI data from the digital drum kit and translated into .csv via the Sekaiju app. Using a complex series of parametric operations, data from the .csv spreadsheet is concatenated and sorted into a series of points along a timeline (Figure 4). From this multiple spatial representations such as solids and meshes (spheres, cones, boxes and more complex forms) velocity and note duration data can be represented symbolically. All parameters are user-adjustable using numeric input panels and sliders. A key attribute of the Grasshopper definition is its flexibility, with parameters such as spatial configuration, bar representation and length, line thickness, note representation, velocity, duration, background and colours all able to be adjusted easily depending on specific user requirements and purpose.

The Grasshopper plugin interfaces with the Rhino3D programme, and all representations can be easily 'baked' (i.e.. transformed from flexible objects derived from Grasshopper to solids, meshes etc. editable in Rhino3D), exported to visualization and animation programmes, brought into Virtual Reality environments, 3D printed, laser cut or fabricated using robotic fabrication processes. A defining element of the research is the mediation between notation, representation and fabrication. The same definition in Grasshopper can be adapted, exported or used in conjunction with other applications to achieve multiple outcomes in the musical, spatial and physical domains.

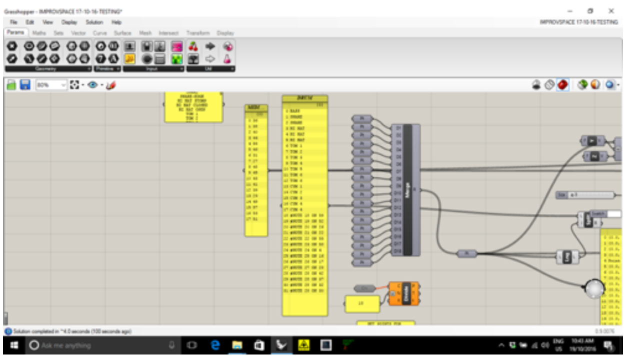


Figure 4. Grasshopper work space showing .csv data being assigned to spatial parameters.

The instrument central to my solo drumming creative practice is a Roland TD20 digital drum kit (see figure 2, above). Although digital, the layout and playing response is very similar to an extended acoustic drum kit. Sounds modeled from real drum kits are enabled in the Digital Audio Workstation through Virtual Instrument applications such as Drumasonic Luxury and BFD3. Virtual instrument software such as BFD stylises the plan form of the drum kit as the basis of their representation of the workspace. This stylised representation recognizes the radial layout that is typical for the drum kit, however the representation is quite literal (Figure 5). The spatial drum notation described below departs from the two-dimensional linearity of traditional notation and the literal representations of the BFD interface to form a symbolic, player-centred representation of the drum kit.

Figure 5. BFD3 virtual instrument interface

The spatial notation schema presented here is the outcome of a design process that attempted to resolve issues inherent in representing the spatiality of the drum kit. The schema stylizes the drum kit in the form of radiating golden section geometric spirals in the X-Y axis (Figure 6), with drum notes represented as events in time along the Z axis. Colour is used as a defining element to enhance the representation of different drums and cymbals. Like all parameters, these are easily changeable. The key to this schema is the practicality of the form as a way of interpreting the drum kit, whilst enabling reading of individual drums from plan and elevation views. The layout is carefully designed to allow the viewing of all drums and cymbals from top, side and bottom planes.

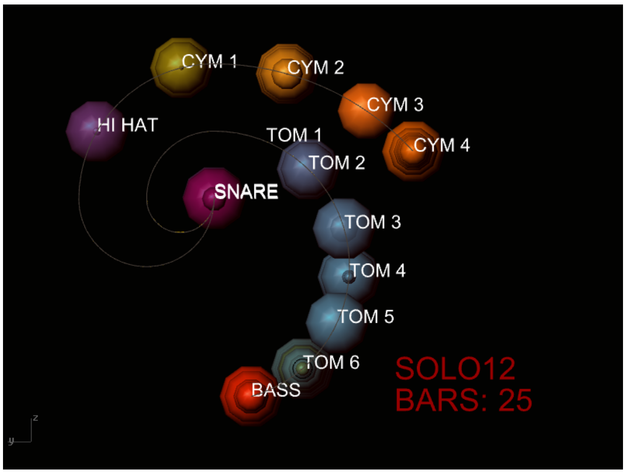


Figure 6. Representational schema for the drum kit with Snare at centre, bass drum at bottom, and hi-hats, cymbals and tom-toms arraying radially.

Whereas traditional notation is designed to be read from one plane only, a key attribute of this spatial drum notation is the capacity to refer to multiple viewpoints in the Rhino3D interface in order to obtain different types of information. The standard Rhino3D interface comprises four viewpoints (plan, two elevations and a 3D perspective or isometric view. Each model view allows the spatial drum notation to be zoomed, panned, rotated and measured by the user to retrieve musical information. As an example, Figure 7 illustrates musical information relating to a drum solo improvisation. This particular improvisation comprises a series of descending roll on tom toms (blue grouped notes) accented by double-kick bass drum notes (red notes) and groupings of hi-hat notes as the timeline progresses. Velocities of drum strikes are represented by the relative diameter of the balls and the time structure is faintly represented as the grid of grey spirals in perspective. Thus, the placement and intensity of notes in the drum solo are given a form and shape within this spatial notation schema and unique elements of style or skill such as rubato can be identified through spatial information contained in this notation.

A second drum improvisation is illustrated in four simultaneous views in the Rhino 3D interface in Figure 8. Each viewpoint provides a different element of this musical information and reading all viewpoints together from the one interface provides a significant body of information that is unavailable in traditional notation. as in architectural representation, certain information can be derived from the plan view (top left), that complements understandings derived from isometric (top right), side elevation (bottom left) and end elevation (bottom right). Through maximizing each view and using zoom, pan and other functions in Rhino 3D, a comprehensive understanding of aspects of drum performance is enabled that is unavailable in other forms of notation.

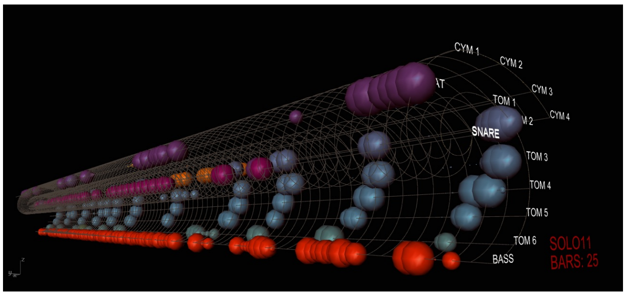


Figure 7. Rhino 3D Representation of a 25 bar drum pattern.

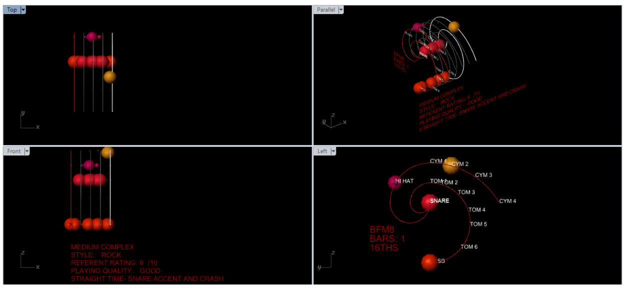


Figure 8. Rhino3D Viewport layout, showing top (plan) view (top left), isometric (top right), side elevation (bottom left) and end elevation (bottom right).

4. DEFINING A LEXICON OF REFERENTS IN 3D SPATIAL NOTATION

'The task of defining improvisation is likely impossible in view of its having no existence outside of its practice (*Brown 2006*)'

Now that the theories and principles of this 3D spatial drum notation have been established, two applications are outlined that are outcomes of creative practice PhD (Architecture and Design) work at RMIT Spatial Information Architecture Laboratory in Australia. A methodology of mass improvisation is used to generate data for the research; to enable reflection on the drumming style developed over thirty years of playing and to form the basis of drum-based compositional works.

The project involved playing a large number of improvisations on the digital drum kit to a basic template of 100 beats-per-minute for one minute, generally at 4/4 time, across three contexts of drumming (playing beats and fills, free form drum solo and playing to a layered guitar track). Playing drums to any template places intrinsic limitation on the outputs and it is recognized that a multitude of different results will be enabled if the tempo, time length or time signature is changed. Given the large body of the author's experience is in the jazz, funk and rock styles this was determined as an appropriate foundation for the research.

Drum improvisations were played on a Roland™ TD20 digital drum kit and recorded in MIDI format on the Reaper™ Digital Audio Workstation (DAW). From the longer one minute, drum solos, beats and tracks, a set of 200 exemplar drum patterns and phrases were extracted in a process of listening, cutting and pasting, and exporting to individual MIDI files. The research draws upon Pressing [26] definition of 'referents' to describe the elements of musical performance that define the player- the 'licks' and 'riffs' that musicians refer to when improvising. These improvisation processes draw parallels with Schon's notion of 'tacit knowing in action [25]'.

A drummer's personal drumming referents are a part of who they are, who their musical identity is. One only has to watch live performances by drummers such as John Bonham to easily identify referent patterns and phrases used, repeated and adapted to different musical situations. The key to this research is to enable a capturing of these referents, using spatial design software to challenge Brown's notion that improvisation has no existence outside of its practice [27].

The author's exemplar set of 200 'referent' drum patterns and phrases were curated, with metadata added in Microsoft Excel™ and Devon Think Pro™ to compliment the sound sample output and spatial notation with identifiers including sample number, a free-word description, complexity of playing, style of music, quality of playing, number of bars, beats per minute and a 'referent rating' that self-evaluated the degree to which they represented the author's individual style. This process has been reported in detail in [28] and [29].

To inform the application of a spatial drum notation, one sample is selected from a much wider body of work (refer figures 9 to 12). Figure 9 provides information about the drums and cymbals used in this sample, including metadata on the sample code and number (SM2) and Number of bars. As all samples for this project were performed at 100 beats per minute, this information is not included, but can be added as required. This information is read directly into Grasshopper from the Excel spreadsheet. In this sample, the snare, bass drum, ride cymbal and Toms 2-6 have been played to produce a pattern described in Figure 10 as ‘snare breakout to toms and bell ending’. This high complexity fill was self-evaluated with a “Referent rating” of 8, where 10/10 is where the patterns absolutely represents the author’s playing style in the area of ‘Rock-Jazz’. The small-diameter spheres on the Bass drum line indicate a series of low-velocity double-kicks whilst the Snare line illustrates a fast set of higher velocity strikes. The sample terminates in a descending Tom Tom roll, with a double strike to the Ride cymbal bell.

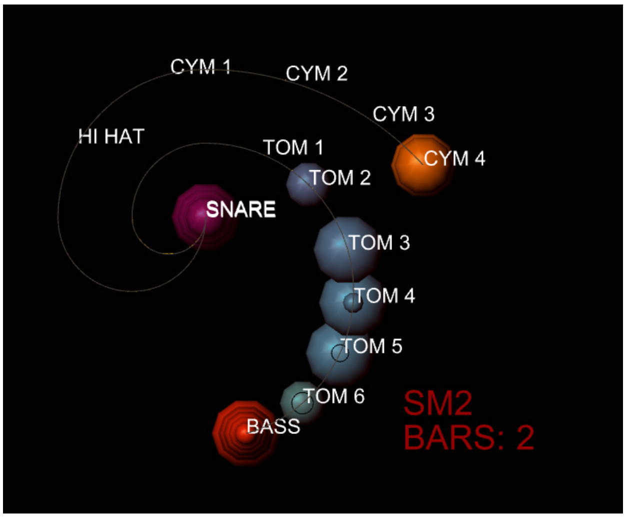


Figure 9. Drum solo Referent No. 2 end view

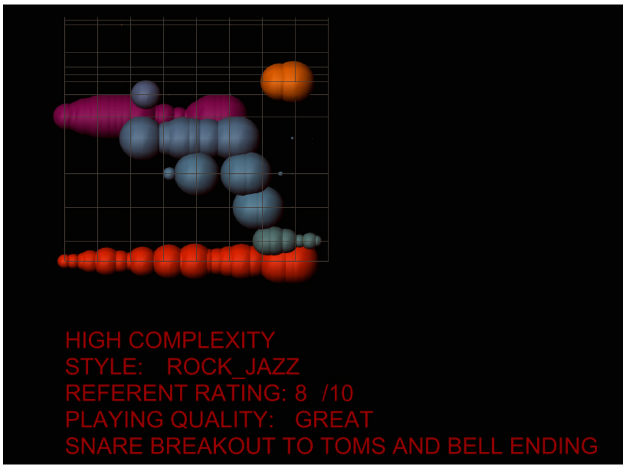


Figure 10. Drum solo Referent No. 2 side view with descriptors on complexity, style, ‘referent rating’, playing quality and free text.

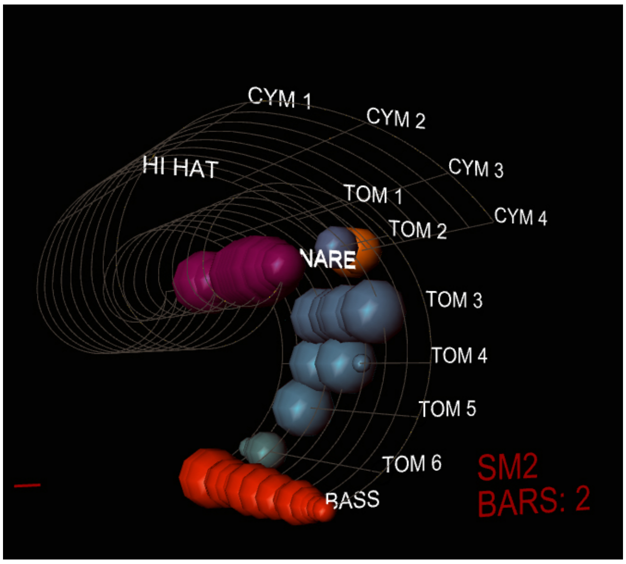


Figure 11. Drum solo Referent No. 2 perspective view

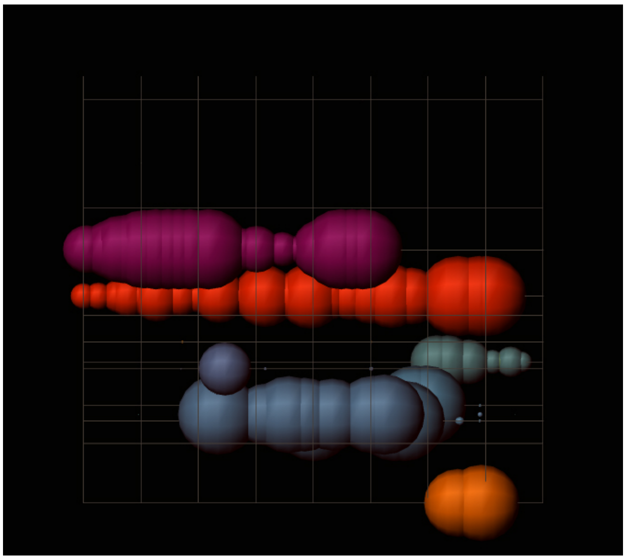


Figure 12. Drum solo Referent No. 2 top (plan) view

As is evident from the four static representations in figures 9-12, a complimentary information set is available from each viewport. Representations in each viewport can be interrogated through zooming, panning, slicing, measuring, animating and other CAD operations. Further, multiple representation options are available to represent the drum events, including using cones (where cone length represents the note duration) and other basic geometries. Sliders in Grasshopper enable quick scrolling through the 200 referents in real-time. It is proposed that this ability to review, analyse and reflect on musical information using multiple spatial representations, colour and through text-based tags significantly overcomes the issues of the musico-spatial perspectival hinge and offers significant affordance for the understanding of the elements of musical drumming style.

These samples, and the metadata schema, provide a large library from which to draw for electronic music composition. The defining characteristic of these samples is the various imperfections, velocity attenuation, microtiming elements and accents that define my style. When looped, sampled and edited using virtual instrument plugins in the Reaper DAW, significant unique compositional opportunities have been made available (see https://soundcloud.com/jjham/). These explorations can occur in both musical and spatial design domains. This initial project has fostered a reflection, using spatial design as the basis, on an established drumming practice. The next stage of the research involves expansions of this practice into diverse domains of music and spatial design. One example is described below.

5. REPRESENTING 'LAYERED RELATIONSHIPS' USING SPATIAL NOTATION

In designing a building, spatial elements (walls, floors, roofs, windows, joinery etc.) are layered on, in and around eachother to form a complete composition in the form of a building. Drawing upon Elizabeth Martin's conception of music and architecture in terms of 'layered relationships [16], the digital drum kit is used to build up a series of improvised layered drum solos to that are layered in, on and around eachother. Whereas for architecture the layering of building elements occurs in space, in this case the layering occurs as drum events in time.

A form of 'spatial polyrhythmic improvisation' is explored on the digital drum kit and represented using 3D spatial drum notation. The principal idea behind this work follows on from view of master drummer Terry Bozzio that the drum kit can be conceived as an orchestra of instruments, rather than a singular instrument itself [30]. By playing with the elements of drumset improvisation identified by Breithaupt [31] (dynamics, tempo, accents, rests, hand to foot distribution and motion) and the 'levers of control' in drum kit practice identified by Bruford [32], the digital drum kit becomes a working tool to enable diverse creative output.

The work also extends the concept of 'drumscapes' coined by David Jones [33] by bringing the drum solo fully into the digital realm. Through sound sampling (including environmental sounds sampled from landscape, the city and buildings), virtual instruments (through Kontakt Massive™, Battery™ and other virtual instrument plugins), spatial polyrhythmic improvisation explores the continuum of music and architecture in both the sonic and spatial dimensions through mixed modalities of improvisation and composition.

'Layered Relationships' is a drum-based composition based on five layers of improvisations from the digital drum kit (see https://soundcloud.com/jjham/). Each layer was recorded on the digital drum kit in the Reaper DAW using a metronome with a visible MIDI piano roll providing visual cues as to the setout of each previously recorded layers. Processed sounds from the Massive synthesizer library were assigned to each layer of drums, with each layer becoming more abstracted and spatial. The compositional intention was thus to build a complex layered drumscape of counterpointed layers of drum improvisations. Although this composition is recorded initially in stereo, research is ongoing in the area of spatial sound, and an adaptation of the spatial notation to incorporate dynamic panning. The five layers of 'Layered Relationships' are illustrated separately in perspective view in Figure 13, below. From this viewpoint, the first few bars of each layer can be viewed providing evidence of the composition of drum selection and relative velocities.

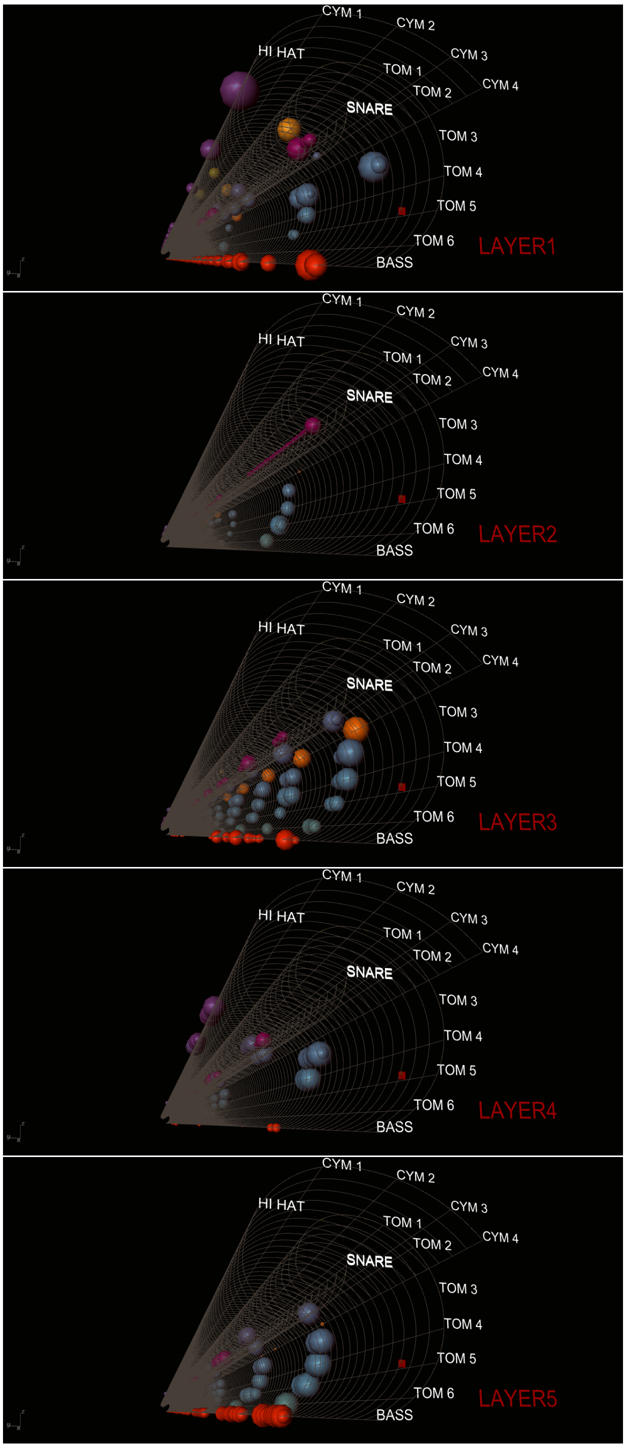


Figure 13. 'Layered Relationships' one point perspective of 5 layers in spatial notation

The layers that constitute a polyrhythmic and multi-layered drum improvisation can be analysed in three dimensions and through multiple viewpoints through spatialized drum notation. Each viewpoint offers different musical information, and all are used to construct complimentary understandings of the composition from a perspective of the notation as three-dimensional spatial elements within a framework of notes and a grid of time.

Referring back to the theory of affordance, spatial notation using CAD tools is particularly helpful in revealing the relationships between layers by using CAD layers to place musical events on. Figures 14 and 15, below illustrate all five layers (represented by different colours) from two different perspectival viewpoints. It is important to note that model views are not static, but can be dynamically panned in and around drum events in time and space. This immersion into, around and through the spatial notation provides opportunities to reveal the detailed interactions between layers, relative velocities, slurs and accents and timing relative to the meter.

Returning to the principles of orthographic projection, 'multiviews' of plan, elevation and section 'help us accurately examine geometric configurations, spatial relationships, and the scale and proportion of a design' for pictorial depth expression, 'single view drawings termed paralines and perspectives are needed [34]. In orthographic representation, certain information is provided on one projection that is complemented by other projections. Together, these projections provide a spatial information set that offers a comprehensive definition of the object under review.

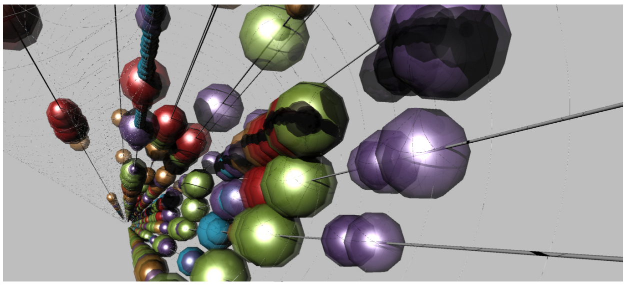


Figure 14. 'Layered Relationships' composition immersive view

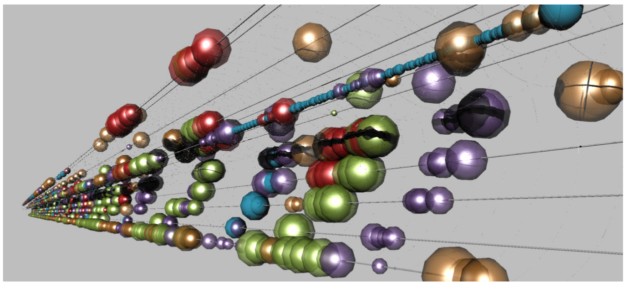


Figure 15. 'Layered Relationships' composition immersive view

6. FROM SPATIAL DRUMS NOTATION TO SPATIALIZATION

Within an integrated 'Musico-Spatial Design' creative practice, opportunities abound for the extension of enquiries beyond the area of spatial notation into the spatialization of drum-based music. The capacity exists to a represent digital drumming improvisations, referent patterns or improvised compositions as notation, 3D spatialization, in Virtual Reality, Augmented Reality and as 3D printed objects and through digital fabrication. Whereas the spatial notation described above utilizes a basic stylization of drum notation, speculative explorations take the creative practice research into more abstract and diverse realms.

Figures 16 and 17 illustrate two different ways of abstracting the composition 'Layered Relationships'. Figure 16 draws on the 'massing representations' used in the field of urban design where clusters of data are brought together to form a representation of the mass of, in this case, drum events in time. By representing in this manner, one can determine the drums most used along the timeline of the composition/ improvisation. In this case, the dominance of use of the snare, bass drum and hi-hats is evidenced by the solid continuous block along the timeline. These clustered representations remove velocity information and thus enable a focus on larger-grained compositional aspects.

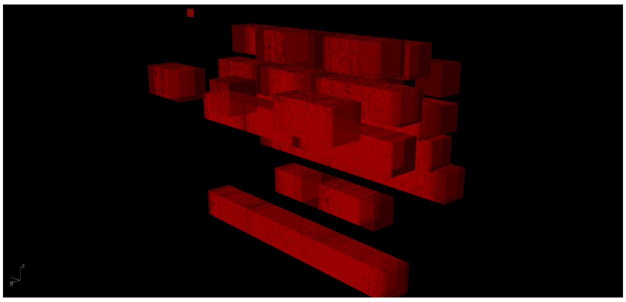


Figure 16. Compacted block representation of 'Layered Relationships'

Extending the abstraction of musical data further, Figure 17 illustrates a lofted representation of the first layer of 'Layered Relationships' and a composite image of all layers (Figure 18). This model was built by setting out drum notes in 3D space (X-Y plane) and velocities in the Z axis. This creates a set of data points in three-dimensional space where a mesh surface can be draped to create a complex curved lofted mesh. As a 3D object, this representation can be exported into different modeling formats for virtual reality, augmented reality, 3D printing and digital fabrication.

Figure 17. Lofted spatial representation of Layer 1 of 'Layered Relationships'

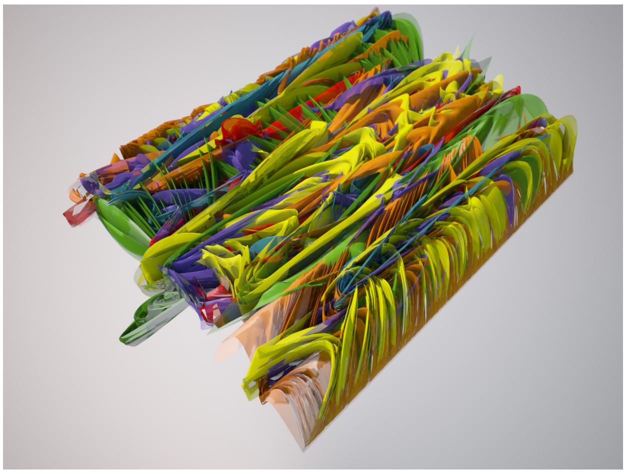


Figure 18. Lofted spatial representation of all layers of 'Layered Relationships'

3D printing potentially extends the affordance of understanding the elements of digital drumming by bringing the notation into being a real object. 3D printed notation allows haptic and physical engagement that overcomes the limitations of the computer screen interface. Although, the concept of 3D printed scores has been explored by Tess [35] to facilitate music reading for sight-impaired people, these 3D printed scores appear to simply operate as a braille form of traditional notation, with raised notes be read by touch. The procedure to build a quality 3D printed score is time-consuming. Drum improvisations and scores written in MIDI are 'baked' in Rhino3D, then exported to Meshmixer and ReplicatorG, then printed with a 3D printer.

Figure 19, below, illustrates an early stage prototype 3D model of a 2-bar drum referent. Longer scores can be built by cutting up a larger model and gluing together. Further potential is enabled by the use of laser cutters or Computer Numerically Controlled (CNC) routers build large scale 2D or 3D musical scores. Work in this area is at the early stages, and ongoing.

Figure 19. 3D printed score of a drum solo referent.

Virtual Reality offers another way in which to afford insights into spatial notation to explore the complexities of polyrhythmic drumming. Working with the University of Stuttgart High Performance Computing Centre Virtual Reality 5-sided CAVE (Cave Automated Virtual Environment), we have conducted early experiments into ways of achieving immersion in spatial notation. VRML models derived from Rhino3D can be 'walked through' using a headset and directional pointer, thus providing a full spatial experience of the score (Figure 19). This overcomes the interface limitations of interrogating spatial notation through the computer screen.

The aim to 3D VR musical notation is noted by Hmeljak [36] to be 'the most intuitive representation of music', and should include 'an appropriate use of symbology and geometry…(and) the use of colours and colour mapping'. We have tested both static 3D VR notation and dynamic 3D VR notation. Whereas static notation is derived from a 'baked' rendered CAD model, the dynamic notation is generated directly from MIDI drum files and, potentially, can be generated through live play in the CAVE. Dynamic notation animates velocities by sending notes into a gravity-simulated virtual environment.

Initial review of this early research work suggests that static notation is a more effective way of enabling reflective understandings of creative practice. Baking and freezing metaphors render the dynamic act of creation into a static object, enabling deeper levels of review and reflection. The dynamic VR system, however, holds much potential for creative practice centred on the spatial design domain. Through our early experiments in Virtual Reality, we affirm the potential uses of the system cited by Hmeljak [36] for computer-aided music analysis, music composition and music education. It is evident that further potential uses are available of CAVE-based drumming in the areas of collaborative CAVE-to-CAVE performance, engagement for non-musicians and exploring an integrated musico-spatial design creative practice wherein the musician-architect is able to create space at the same time as creating music.

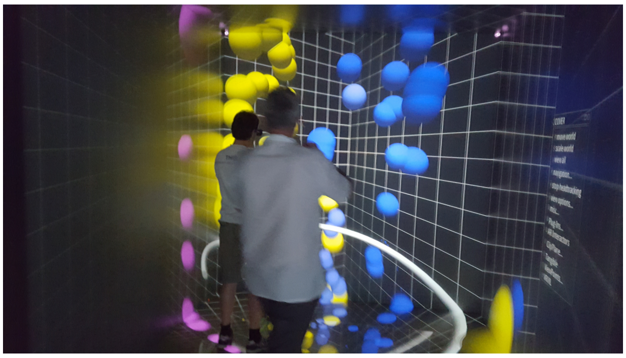


Figure 20. Dynamic VR inside the CAVE.

7. CONCLUSIONS AND FUTURE RESEARCH

This research presents a framework for spatializing drum notation founded on the principles and theories of architectural representation using the tools of architectural practice. The creative research project work presented here: the reflection on a lexicon of referent drum patterns and phrases; a multi-layered 'drumscape' composition and extension beyond notation into abstracted representations, 3D printed scores and into Virtual Reality environments are intended to demonstrate the potential of that the domains of architecture and spatial design can bring to that of musical notation.

Clearly, this research offers potentials that compliment or extend traditional notation and the area of 3D notation. The point of interest is how this spatial notation system affords insights into the complexities of polyrhythmic playing, and enables a spatial mapping and representation of the elements of individual drummer's musical styles. The focus of this paper has been the author's reflective and compositional work and the development of a system primarily intended for internal use. If a theoretical 'musico-perspectival hinge' indeed exists and notation that is limited to two dimensions offers less affordance of musical knowledge, this spatial notation system may provide the basis for many further explorations.

One such exploration that will be reported in future research is the use of spatial notation to compare the improvisation of different drummers improvising over the same 'template'. A reversal of the Grasshopper definition is underway that will allow the exploration of composition using spatial notational, with resultant output in MIDI format. These are just some of the fertile areas available for exploration within an integrated musico-spatial design creative practice.

Acknowledgments

This research is supported by an Australian Government Postgraduate Research Scholarship. The author would like to acknowledge Daniel Prohasky and Awnili Shabnam for their fine work on the Grasshopper script and Uwe Woessner and Joachim Kieferle for their work inside the CAVE.

8. REFERENCES

1. L. Kalisperis and A. Pehlivanidou-Liakata. *Computers in Design Studio Teaching'*. in *EAAE-eCAADe International Workshop Architectural Design Studio: Digital and Traditional*. 1998.
2. J. Bermudez and K. King, *Media interaction and design process: Establishing a knowledge base.* Automation in Construction, 2000. **9**(1): p. 37-56.
3. A. Pérez-Gómez and L. Pelletier, *Architectural Representation and the Perspective Hinge. Cambridge, Massachusetts & London*. 1997, The MIT Press.
4. J. J. Ham, *On Representation and Construction in a Tectonic Design Studio*, in *School of Architecture and Building*. 2004, Deakin University: Geelong, Australia.
5. K. Stone,, *Music notation in the twentieth century: a practical guidebook*. 1980: WW Norton.
6. J. Gibson, *The Theory of Affordances in the Ecological Approach to Visual Perceptual*. 1979, Houghton Mifflin.
7. D. Norman, *The design of everyday things: Revised and expanded edition*. 2013: Basic books.
8. N. Weinberg, *Guidelines for Drumset Notation.* Percussive Notes, 1994: p. 15.
9. M. Muecke and M.S. Zach, *Essays on the Intersection of Music and Architecture*. Vol. 1. 2007: Lulu. com.
10. N. Goodman, *Languages of art: An approach to a theory of symbols*. 1968: Hackett publishing.
11. C. MacNaughtan, *Indeterminate notation and sound in the space of architecture.* The Journal of Architecture, 2006. **11**(3): p. 335-344.
12. P. Rebelo, *Notating the unpredictable.* Contemporary Music Review, 2010. **29**(1): p. 17-27.
13. J. Cage, *Notations*. 1969, New York: Something Else Press.
14. C. Hope, L. Vickery, A. Wyatt and S. James, *The Decibel scoreplayer–a digital tool for reading graphic notation*. in *International conference on technologies for music notation and representation (TENOR 2015). IRCAM, Paris*. 2015.
15. R. Wittkower, *Architectural principles in the age of humanism*. 1971: WW Norton & Company.
16. E. Martin, *Pamphlet Architecture 16: Architecture as a translation of music*. 1994: Princeton Architectural Press.
17. Lab-au.com. *The shape of sound*. 2015 [accessed 24th April 2015]; Available from: http://lab-au.com/theory/article_soundscapes/.
18. A. J. Levy, *Real and Virtual Spaces Generated By Music.* International Journal of Architectural Computing, 2003. **1**(3): p. 375-391.
19. M. McCullough, *Abstracting craft: The practiced digital hand*. 1998: MIT press.
20. P. Ferschin, D. Lehner, and Y. Oka, *Translating music into architecture*, in *The third International Mathematics and Design Conference M&D2001: digital, hand, eye, ear, mind*. 2001, Mathematics & Design Association: Deakin University Geelong, Australia.
21. P. Christensen and M. A. Schnabel, *Spatial Polyphony: Virtual Architecture Generated from the Music of J.S. Bach*, in *CAADRIA 2008 [Proceedings of the 13th International Conference on Computer Aided Architectural Design Research in Asia]* 2008: Chiang Mai (Thailand). p. 501-509.
22. R. J. Krawczyk, *Exploring the Visualization of Music*. in *Proceedings of Bridges 2012: Mathematics, Music, Art, Architecture, Culture*. 2012. Tessellations Publishing.
23. M. Novak, *Liquid Architectures in Cyberspace (1991).* Multimedia: From Wagner to Virtual Reality, 2002: p. 272.
24. J. H. Hansen, *Musical Sculptures*, 2012: [accessed 24th April 2015];

https://www.youtube.com/watch?v=uxYHlZQSADQ&t=71.
25. D. Schön, *The reflective practitioner: How professionals think in action*. Vol. 5126. 1983: Basic books.
26. J. Pressing, *Improvisation: methods and models.* John A. Sloboda (Hg.): Generative processes in music, Oxford, 1987: p. 129-178.
27. D. Brown, *Noise orders : jazz, improvisation, and architecture*. 2006, Minneapolis: University of Minnesota Press. 159 p.
28. J.J. Ham and D. Prohasky, *Developing a parametric spatial framework for digital drumming*, in *13th Sound and Music Computing Conference*, R.G.a.G. Hajdu, Editor. 2016, Zentrum für Mikrotonale Musik und Multimediale Komposition, Hochschule für Musik und Theater Hamburg Germany. p. 197-202.
29. J. Ham, M. A. Schnabel, L. Harvey and D. Prohasky, *'Starting with Nothing' and 'Ending up with Something' - Musical Improvisation and Parametric Spatial Design Improvisation*, in *Complexity & Simplicity - Proceedings of the 34th eCAADe Conference, A.* Herneoja, Editor. 2016: University of Oulu, Oulu, Finland. p. 377-386
30. D. Zulaica, *Terry Bozzio: An Interview From The Vault Part I*. 2013 [accessed 2015 29th March]; Available from: http://www.drummagazine.com/features/post/terry-bozzio-interview-from-the-vault-part-i/.
31. R. Breithaupt, *Musical Considerations for Drumset Improvisation.* Percussive Notes, 1987. **26**(1): p. 15-16.
32. W. Bruford, *Making it Work: Creative music performance and the Western kit drummer*, in *School of Arts, Faculty of Arts and Social Sciences* 2015, University of Surrey: Surrey UK.
33. D. Jones, *Drumscapes Solo Concert recorded at Melbourne Recital Centre*. 2013, Independent Release: Melbourne.
34. R. Yee, *Architectural drawing: a visual compendium of types and methods*. 2012: John Wiley & Sons.
35. P. Tess. *3D printed Tactile Stave Notation helps visually impaired to better read music notations*. 2015 Nov 9, 2015 [accessed 2016 30th September 2016]; Available from: http://www.3ders.org/articles/20151109-3d-printed-music-notation-system-for-the-visually-impaired.html.
36. D. Hmeljak, *3D Musical Notation-Providing Multiple Visual Cues For Musical Analysis*, in *Unknown*. Unknown.

A CAP for graphic scores
Graphic notation and performance

Benny Sluchin
EIC-Ircam
sluchin@ircam.fr

Mikhail Malt
Ircam
Mikhail.Malt@ircam.fr

ABSTRACT

Many graphic scores use the pitch versus time presentation, as a natural extension of the usual notation. In the general case, it displays discrete pitches, in a fixed timeline. Nevertheless, graphic scores use a lot of continuous lines, and the vertical dimension can be adapted to a particular performance. In such a way, the instrumentation is free, and the actual range of a particular instrument can be adapted according to the notation. The present article is initiated by a search to provide the performer with adequate tools to approach the execution of such works. A computer assisted performance approach helps the player in the preparation process for both: the time and the pitch approximations. The simulation can enhance the performance in approaching the graphical notation.

.

1. INTRODUCTION

The starting point of the present research was a performance issue; works of the 20th century, aimed at concerts and recordings, were approached seriously. The player searched for tools that will help him attain the composer's intentions as proposed by the printed (graphical) scores. Curiously enough, such tools did not exist, and the precision of the performance was quite lacking. Our work focuses on this category of works and a CAP (Computer Assisted Performance) approach, that is in fact help in preparing the works for performance, in the same way a metronome is.

2. PREVIOUS WORK

Since several years now, works having a generic presentation, that is to say, open form, has occupied us. Three examples were discussed: *Domaines* (1968) by Pierre Boulez, *Duel* (1959) and Strategy (1962) and *Linaia Agon* by Iannis Xenakis and *Concert for Piano and Orchestra* (1958) by John Cage ([1], [2], [3]). Other works by Cage, part of his late "number pieces" were also accessed [4]. In all these cases, a computer interface that assist the performer in preparing and performing the works were developed.

The two main compositions we discuss here, present performance notation difficulties. Conversely to previous work [5][1], our aim is to develop solutions, in a computer-assisted performance study, for trombone version, and offer aural guides to attain a performance, as close as possible to the graphical notations used.

3. TWO EXAMPLES, TWO PROBLEMS

3.1 Alvin Lucier's *Panorama* (1993)

Scored for Trombone and piano, *Panorama* (1993) [6] is a work whose shape is traced by the sliding trombone part (Figure 1).

"During the course of the work, the trombonist slides continuously, outlining the shapes drawn by the diagonal lines." [6]

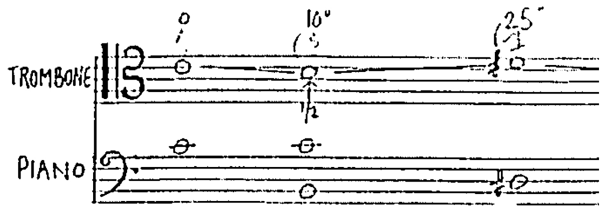


Figure 1: Alvin Lucier's *Panorama*, original manuscript score, first 25 seconds

The musician may stop sounding his instrument at any time but his slide moves continuously. The pitches are notated with great precision within microtonal tunings. Alvin Lucier uses a half, quarter, third and sixth note notation he also adds a shift notation, in cycles on some pitches (Figure 2) and give exact onset time (minutes and seconds).

As quoted by Volker Straebel [7], "Alvin Lucier mapped the panorama of the Swiss Alps to the pitches of a slide trombone. He worked from a reproduction of a panorama drawing by Fritz Morach after a landscape photo by Hermann Vögeli".

[1] Enrico Francioni develops a technical solution for one-man *Ryoanji* performance, in the contrabass, percussion and vocalize version.

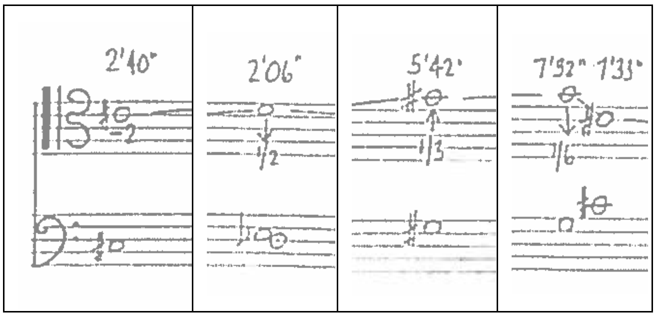


Figure 2: Alvin Lucier's Panorama, microtonal notation

3.2 John Cage's *Ryoanji* (1983-84)

Ryoanji (1983-4) [8] is a series of works based on the celebrated rock gardens in Kyoto. The contours of the fifteen different rocks are the basis of the graphic notation. J. Cage's instructions are typical; they give the poetic atmosphere for the performance: "*Each two pages are a 'garden' of sounds. The glissandi are to be played smoothly and as much as is possible like sound events in nature rather than sounds in music. (...) The score is a 'still' photograph of mobile circumstances.*" [8]

The solo instrument parts (there are versions for oboe, flute, trombone voice and double bass) appear all in the same form. The original score presents a frame where continuous and complex gestures were drawn. For each page, in the left upper part, we can remark a small stave with the page register (Figure 3). On the abscissa, the time (the length of a system) is represented on 24 cm. The height, the register used, is represented on 6 cm and subdivided, according to the register, in semitones.

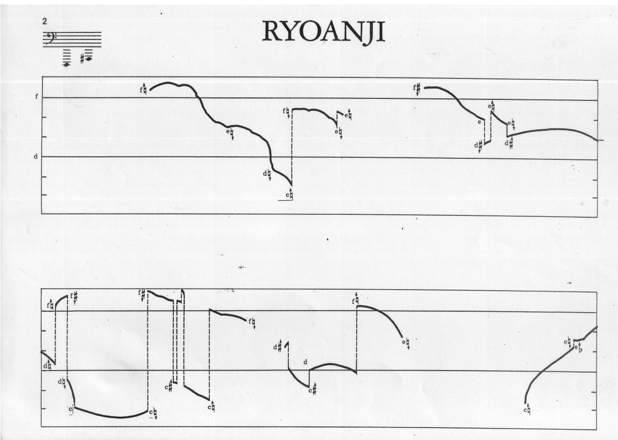


Figure 3: Cage's Ryoanji first trombone version page

In Figure 3 the range is precise (from B0 to F#1), and the durations are to be determined by the performer. Concerning the duration, only in the voice version Cage was explicit (ca. 2 minutes for a "garden", composed of four systems, engraved on two pages). For the instrumental versions, musicians generally apply this same duration. Each "garden" is using a specific range, which is shown in Figure 4.

The vertical span (which is the same for all the pieces) indicates a variety of intervals going from a semi tone (J3 in the flute version, J6 in the double bass version) to a twelfth (J8 in the oboe version).

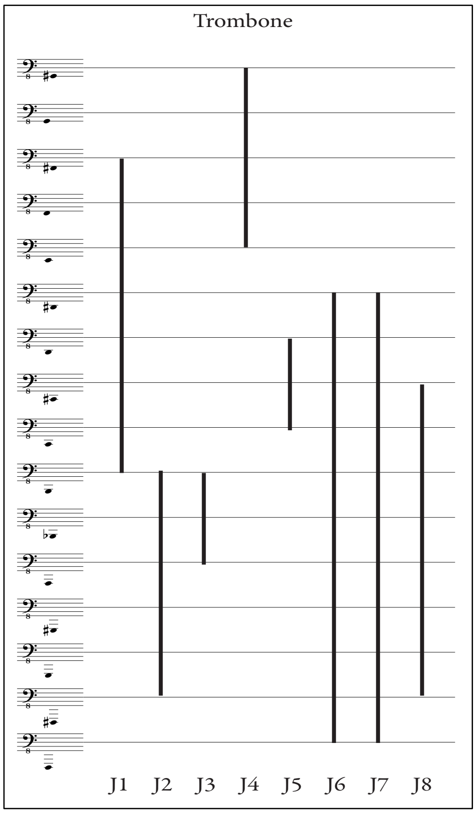


Figure 4: Ryoanji Trombone Gardens Registers

The form of the curves is very precise, but hard to perform on the instruments. Both, the time factor (horizontal abscissa) and pitches (vertical) are to be approximated with maximal precision.

4. THE PERFORMANCE ISSUE

4.1 Pitch discrimination threshold

According to some authors ([9], [10]) the pitch interval discrimination for trained musicians is around a semitone tenth or 10 cents at 500 Hz (around B4) and increases to lower pitches (100 Hz), until a half-tone. Two main aspects present difficulties in these pieces. The first is the difficulty to control slow glissando, where the discrimination of pitch change is challenging, as in *Panorama*. Lucier himself says:

> "The player is not expected to be able to hear these finer tunings, but may perceive them as beats per seconds against the piano tones" [6].

The second, mainly related to *Ryoanji*, is to evolve in a very low instrumental register. *Ryoanji*, trombone version, evolves between F-1 (21,83 Hz) and G#0 (51,91 Hz). In this register the pitch discrimination is far to enable the fine-tuning of the notated glissando curves. In addition, the performer physical control, lips vibration and slide position, is difficult and uncertain.

4.2 The Glissando

Defined as a glide from one pitch to another, the term contains much more than the physical change of the frequency. There are musical examples (ethno-musical and contemporary) based exclusively on it. Its musical performance, even on very adapted instruments (i.e. unfretted bowed strings, slide trombone), is far from simple. Traditional orchestration books, that discuss glissandi, miss a practical approach which therefore follows.

A glissando has two primary features: the start and ending points, and the duration. A secondary character, the form of the sliding is particularly important, can be defined as the speed. A glissando can be linear (having a constant speed) or not linear (having a varying speed). Performing a glissando on a trombone[2] (the instrument related to this research) is strongly related to the register involved. For middle and upper registers the execution is determined by the slide position taken as a starting point. Naturally, one will be unable to perform an ascending glissando if the starting slide position has the shortest slide length. Alternate slide positions for the same note are then needed. Another problem, concerning the doability of a glissando, is related to the fact that a trombone slide can execute a maximum interval of an augmented fourth (the span of seven positions a Bb trombone slide has). In many cases, a glissando has to be tiled from two or more simple ones.

The slide length modification needed to perform a glissando may be of some help to measure the speed of the glissando. The modification to perform a given interval changes according to the trombone position. For example, an interval of a half tone needs a greater slide length modification in lower positions than in higher ones.

The slide is not the only way to control the pitch. Particularly in the low register, the lips can operate important frequency changes. These changes are though accompanied by a loss of the particular timbre. As the lips are vibrating in a different frequency from that corresponding to the instrument length.

When performing a glissando, the starting pitch is often just a floating moment. The pitch has to start moving immediately, though the ear won't perceive a change initially. The same holds for the ending point of a glissando.

Another way to measure a glissando, lies on the notion of differential beats. Having a reference pitch, close in frequency to the moving one, will create this phenomenon. Controlling the relationship between two close frequencies, based on the number of differential beats in time, is an approximate and quite effective method.

[2] It seems that some of these points may be generalized to string instruments.

5. COMPUTER SIMULATION

As pointed previously the performance of Alvin's Lucier *Panorama* and Cage's *Ryoanji* presents some difficulties mainly related to glissandi control, pitch discrimination and extreme low pitch control. The leading idea using simulation was to produce an exact and precise guide for a performance study.

5.1 Alvin Lucier's *Panorama* (1993

Before building the computer simulation a first step was intended. *Panorama* was transcribed in a Cartesian space, in a break point function presentation. Each page is having the same duration of one minute (Figure 5).

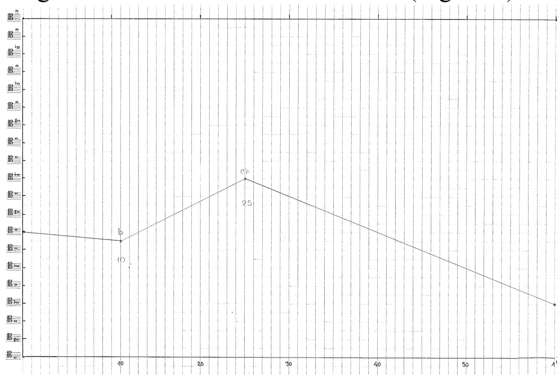


Figure 5: Graphic score containing pitches

This representation brought a better score awareness of both pitch versus time evolution.

The second attempt, to improve and help performance, was to transcribe the original manuscript (Figure 1) in a time proportional musical notation score. In the original there is not a clear correlation between time and graphic space. Each system lasts for one minute (Figure 6) and the dashed measure bars represent regular 10 seconds time span.

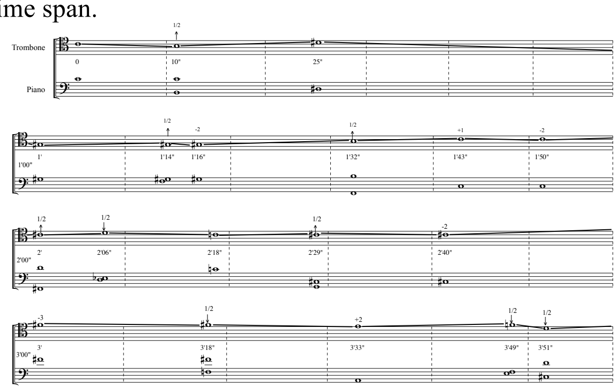


Figure 6: Alvin Lucier's *Panorama*, transcribed score, first page

Even if it could seem minor, it is not. This transcription improves the readability over the manuscript and a good time versus graphic space proportionality helps the performer to better evaluate the linear time span.

Controlling the slow pitch change over time is one main difficulties for *Panorama*. For example, at 1'43", the trombone evolves over 7 seconds with a pitch change from a C4 plus 1 Hz to a C4 minus 2 Hz. This change only can be made (as in Xenakis pieces) by referring to

the differential beats produced between the trombone C4 fundamental and the piano C3 octave harmonic. But it still presents a 0.4286 Hz/sec variation. Another moment, from 14’02” to 14’14”, presents a 0.1667 Hz/sec variation on a Bb3. In general, the pitch slope is very small, in relation to the pitch discrimination interval. In addition to these four crucial points (1’43”, 6’12”, 9’37” and 14’02”), the piece presents other problematic pitch slope variations. For Example, the smallest slope is to be found from 7’05” to 7’32”, where the performer is asked to control a descending glissando (from a G4 to a G4 minus a sixth tone), over 27 seconds. A 0.59 cents/sec variation.
These points lead us to provide a simulation to help the performer with the tuning and the duration. Event if the pitch changes is still below the pitch discrimination hearing range. The performer, playing together with the simulation, can refer to the beating as a way to adjust the pitch.
From graphical data inferred in the precedent steps, a three columns table with time (in seconds), MIDI pitch and the shift in Hz (Figure 7) was created.

0	60	0
10	59.5	0
25	63	0
60	56	0
74	55.5	0
76	56	-2
92	59.5	0

Figure 7: First seven trombone events (time in sec, MIDI pitch and shift in Hertz)

Plotting the table data from Figure 7 we obtain a glissando curve displaying the Alps landscape (Figure 8) [7].

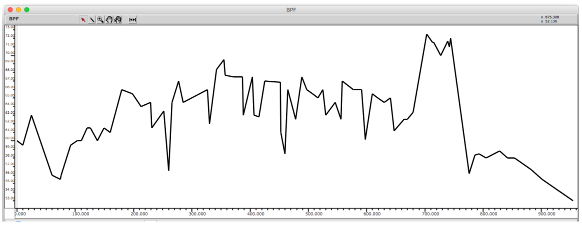


Figure 8: Alvin Lucier’s *Panorama* glissando curve (Time x MIDI pitch)

The synthesis simulation was done with Csound[3] in the OpenMusic environment[4]. The Csound orchestra was simplest as possible (Figure 9) just a single oscillator (<oscili>) where the frequency (k1), is driven by a <f2> score function[5].

```
Instr 1
k1  oscil1  0, 1.0 ,p3, 2
asig  oscili  p4,k1,1
    out  asig
```

Figure 9: Csound orchestra for *Panorama* synthesis simulation

The <f2> score function is built from the table data (Figure 8). And the spectral content is set via a <f1> table in order to have a 5 partials harmonic sound (Figure 10)

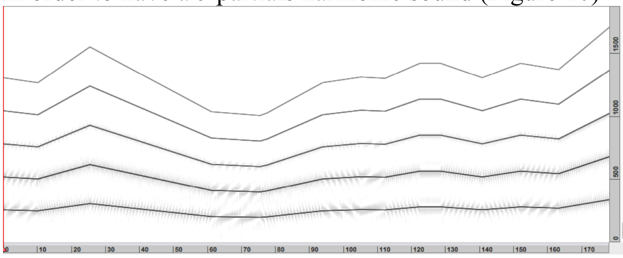


Figure 10: Alvin Lucier’s *Panorama*, first 3minutes, 5 harmonic sound sonogram

As the aim of the simulation is to be a guide to a performance study, the use of a 5 partials harmonic sound is intended to help the performer to tune and to follow more precisely the glissando. Beating between higher partials are more perceptible than beating between lower partials.
Another simulation performance guide was built using a sample of Eb4 ordinario trombone sound (from the SOL sound, Ircam data base). This 4 seconds real sound was stretched until to reach 16’40” using Audiosculpt (© Ircam)[6]. The stretched sound was then continuously transposed using the data from Figure 7, with the SuperVP Kernel (© Ircam)[7]. As the mean pitch from all the performance is 63.31 MIDI pitch and the median 63.24 MIDI, we choose Eb4 to minimize the transpositions of the original sound, keeping as possible its concrete character.
A Max (© Cycling74)[8] patch was built then, to allow the performer to browse in the simulation to use the transformed sound as a guide to study (Figure 11).

5.2 John Cage’s *Ryoanji* (1983–84)

For *Ryoanji*, the main idea was, likely to *Panorama*, to build a simulation, a guide to be used in the study performance phase, but the basic approach was completely different. The glissandi here are not linear but freely drawn by Cage.
The first step was to find a strategy to retrieve the pitch versus time information from Cage score.

Some strategies were studied, like border detection in the picture file, as a methodology to identify the curve. But for all of them the preprocessing step (cleaning the original score, erasing the frames, etc.) was to much time consuming. Instead of that we adopted an old computer technic, “hand digitalization of plotted data” (as done in

[3] http://www.csounds.com
[4] With two external Libraries, OM2Csound and Chroma. See: http://repmus.ircam.fr/openmusic/libraries. For OpenMusic information: http://repmus.ircam.fr/openmusic/home
[5] This table, <f2>, is indicated as being the last <oscili> parameter (2).

[6] http://anasynth.ircam.fr/home/english/software/audiosculpt
[7] http://anasynth.ircam.fr/home/english/software/supervp
[8] https://cycling74.com/

the 70's). This technic involves having a background image and with another tool to identify points (coordinates) in the background image. It was broadly used in sea waves digitalization.

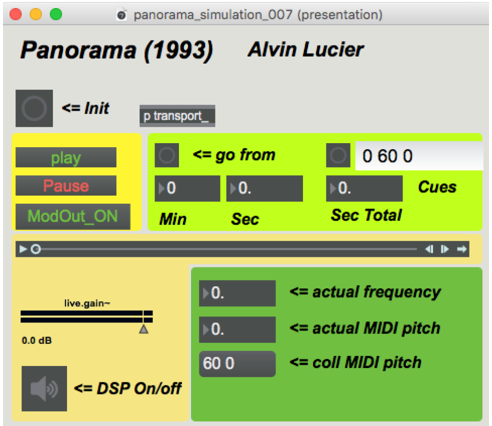


Figure 11: Computer interface to browse *Panorama* simulation

Using the OpenMusic (© Ircam) environment, we firstly reconstructed the scanned score to have each garden on one linear page to proceed afterwards to hand digitalization.
As the original score was scanned from a manuscript, all the frames, from the same garden, has not exactly the same height. The bottom and upper border were not exactly parallel and the semi-tone subdivisions were not aligned across all the garden frames. To say that the alignment was not so easy (Figure 12).

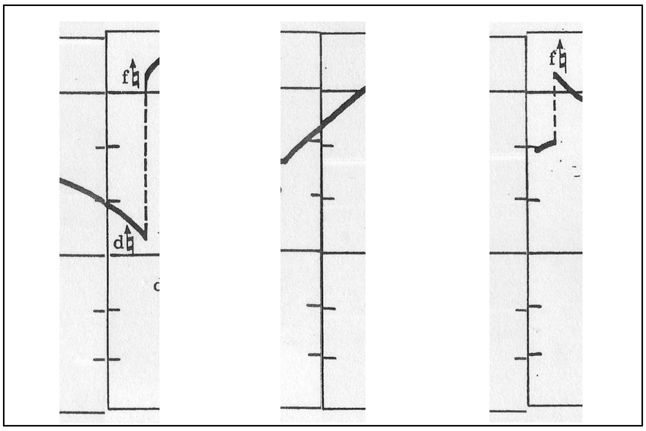


Figure 12: Tree intersections, from Cage's *Ryoanji* first trombone garden

To resolve this drawback, we chose to align the different frames focusing on the continuity of the line describing the gesture (Figure 13).
From the reconstructed garden, we proceeded to "hand digitalization", with the help of the <bpf> OpenMusic object (Figure 14), trying to work as precisely the zoom allow us to do. Compare with curves in Figure 3.

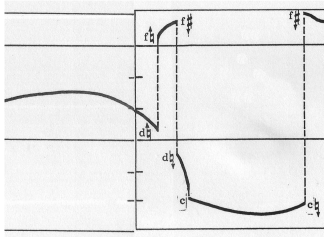


Figure 13: From first to second frame, Cage's *Ryoanji*, first trombone garden

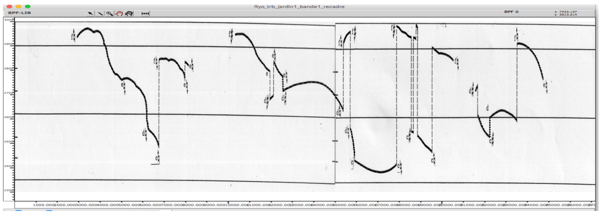


Figure 14: Digitalized curves, inside OpenMusic <bpf> object

Following the curves digitalized (Figure 15), it was easy to isolate them and use each one as a frequency data curve to a Csound synthesizer as used previously for *Panorama* simulation.

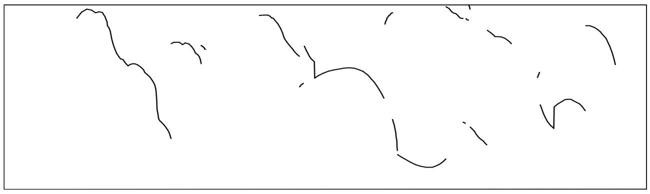


Figure 15: Extracted glissando curves

With this last figure one can imagine the stone outlines, Cage used for the score generation [12].
While the registers are given, the time scale is up to the performer. We choose to give to each garden a two-minute duration.
All the gestures were synthesized as for *Panorama*, but as the fundamental frequencies were to low we used a seven harmonics spectra to help the tuning (Figure 16).
The final simulation sound file, would be imported in an interface (as for Panorama), to allow the performer to browse and use it for preparing his performance.

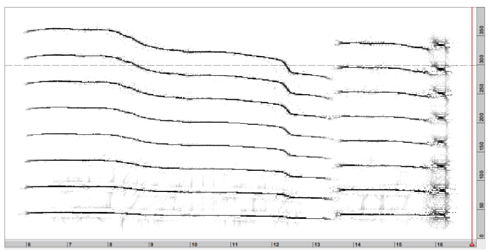


Figure 16: *Ryoanji*, first trombone garden, first 16 sec sonogram

6. CONCLUSIONS

The two pieces treated here take advantage of graphical notation but present essential difficulties in performance. We propose here a computer strategy, simulation based, to assist the musician in the preparation and performance.
Concerning *Panorama*, while we found a way to bring a possible assistance for the glissando control and fine-tuning, the optimization for the glissandi position changes still is to do. We look forward to deal with it mainly with a constraint system. Concerning *Ryoanji*, the main problem (glissando control and fine-tuning) still unsolved. As pointed previously, main parts from this work still impracticable, from cognitive and instrumental reasons. *Ryoanji* evolves in the extreme lower register what makes very difficult to control and listen to the small microtonal waves. For the first garden (where Cage's register is from B1 to F#1) the gestures (according with our hand digitalization) evolve between C1 minus 19 cents (32.35 Hz) and a F#1 minus 14 cents (45.88 Hz).
We have two pieces mainly based on glissandi, but with two different performance goals. While in Panorama Lucier expects for a precise pitch and time performance, in *Ryoanji*, the graphical notation has not only a score function, but also an inspirational and poetic purpose. As stated in the *Ryoanji* instructions:

> "The glissandi are to be played smoothly and much as is possible like sound events in nature rather than sounds in music." [8]

Composers, inspired by extra musical facts or concepts can use sonification as an inspiration way. Translating graphic or visual aspects in musical notation can be source of novelty and challenge, but sometimes it can be source of difficulty.
The notation graphic space does not have the same properties as the basic pitch versus time real space. If in the graphic space, we can represent any pitch interval size, it is just a scale question dependence. In hearing world, humans are bounded by their cognitive pitch thresholds. While in the graphic space, we can navigate in the time line, jumping from future to past and vice-versa, in the real world the time appears to be irreversible.
These two musical compositions were recorded based upon the present research [14].

7. REFERENCES

[1] B. Sluchin, M. Malt, "Open form and two combinatorial musical models: The cases of Domaines and Duel", *3rd International Conference on Mathematics and Computation in Music*, Paris, June 15-17, 2011, p. 211-217.

[2] M. Malt, B. Sluchin, "Théorie des jeux et structure formelle dans Duel (1959) et Stratégie (1962) de Xenakis", *L'influence des théories scientifiques sur le renouvellement des formes musicales dans la musique contemporaine*, Symposium, organized by M. Grabócz, CDMC, Paris, 28 november 2014. https://www.youtube.com/watch?v=ZixIYc7iL84&feature=youtube_gdata

[3] B. Sluchin, "Linaia-Agon Towards an Interpretation Based on the Theory", *International Symposium Iannis Xenakis*, M. Solomos, A. Georgaki, G. Zervos (Ed.), Athens, May 2005, 1–16. http://Cicm.Mshparisnord.org/ColloqueXenakis/

[4] B. Sluchin, M. Malt, "Interpretation and Computer Assistance in John Cage's Concert for Piano and Orchestra (1957-58)", *SMC - Sound and Music Computing Conference*, 2010, pp. 1 – 8.

[5] E. Francioni, "Csound for Cage's Ryoanji", *Csound Journal*, vol. 18, 2014, 1 – 20. http://www.csounds.com/journal/issue18/francioni.html

[6] A. Lucier, *Panorama for trombone and piano*, manuscript score, 1993.

[7] V. Straebel, "The Sonification Metaphor in Instrumental Music and Sonification's Romantic Implications", *The 16th International Conference on Auditory Display (ICAD-2010)*, 2010, pp. 1 – 8.

[8] J. Cage, *Ryoanji for Trombone Solo*, Henmar Press Inc., 1985.

[9] J. H. McDermott, M. V. Keebler, C. Micheyl, A. J. Oxenham, "Musical intervals and relative pitch: Frequency resolution, not interval resolution, is special" *The Journal of the Acoustical Society of America*, vol. 128, no 4, 2010, pp. 1943 – 1951.

[10] F. Bianchi, S. Santurette, D. Wendt, T. Dau, "Pitch discrimination in musicians and non-musicians: Effects of harmonic resolvability and processing effort" *Journal of the Association for Research in Otolaryngology: JARO*, vol. 17, 2016, pp. 69 – 79.

[11] J. Cage, "Ryoanji: Solos for Oboe, Flute, Contrabass, Voice, Trombone with Percussion or Orchestral Obbligato (1983-85)", *PAJ: a Journal of Performance and Art*, vol. 31, no 3, 2009, pp. 58 – 64.

[12] M. Fowler, "Finding Cage at Ryōan-ji Through a Re-Modelling of Variations II." *Perspectives of New Music*, vol. 47, no 1, Winter, 2009, pp. 174 – 191.

[13] K. Schöning, "Ryoân-ji: oder die Gärten der Leere", *Neue Zeitschrift Für Musik (1991-)*, vol. 155, no 5, 1994, pp. 24 – 27.

[14] J. Cage, *Ryoanji*, Benny Sluchin – trombone, Sylvain Rappaport – percussion, CD produced by ut_performance@hotmail.com, 2015, Paris, France.

ARE SCORES MAPS? A CARTOGRAPHIC RESPONSE TO GOODMAN

Daniel Miller
Dartmouth College
daniel.j.miller.gr@dartmouth.edu

ABSTRACT

Nelson Goodman's theory of notation attempts to provide an ambitious, unified account of how systems of symbolic representation preserve and transmit information and how they differ from pictorial depiction. However, Goodman's account of music and dance notation has proven unpopular, with some critics objecting to the rigor with which scores and musical symbols are assumed to designate musical works and their constituent elements. This paper reconsiders a Goodmanian account of a music notation system in the light of recent philosophical work on maps and map-like cognition. Specifically, I propose that scores do not act as compound symbols that uniquely designate musical works. Instead notational components of scores are better understood as contingent surface-level features leveraged by an underlying map-like representational structure. On this account, scores are seen to be highly conventionalized maps, and the notational symbols of scores constitute just one of multiple modes of representation and depiction harnessed by this framework. Finally, I consider several contemporary examples of music notation and discuss how a cartographic theory of notation may provide novel insights into the graphic design considerations of these scores, particularly those that rely on new notation platforms such as graphic design software or animation, where depictive and symbolic strategies are frequently hybridized.

1. INTRODUCTION

Music in the European tradition has frequently been compared with language, and insofar as the score and notation are assumed to be the primary communicative vehicle of a work of music or dance, scores have also been compared to the written word. Nelson Goodman's theory of notation represents a highly refined version of this argument; music notations are analyzed as having the same form and function as the symbolic representations of languages. In this paper I propose that a cartographic system of representation is a plausible alternative to a sentential theory of music scores. I follow Elisabeth Camp, who has argued on both philosophical and neurological grounds that a map-like form of cognition is an alternative model to the "Language of Thought" argument, which holds that thought must be language-like [1]. In what follows, I will first summarize Goodman's theory of notational scores and objections to the theory. I will then provide an outline for a cartographic theory of scores (although space does not permit a complete exposition of the argument), and I will conclude with several practical examples of score design problems that might benefit from an analysis of the logic of and graphic design in maps.

1.1 Languages of Art

Nelson Goodman's theory of music notation arises from his broader interest in symbols, which is mainly set forth in his 1968 book, *Languages of Art*. Subtitled "An Approach to a General Theory of Symbols," this ambitious project sought to establish a unified analytic theory of symbols that would be broad enough to encompass the many disciplines in which they function, including natural languages, visual arts, music, dance, and the sciences.

Although appealing in its scope and explanatory power, Goodman's project has been unpopular with philosophers as a theory of music or dance notation and has been largely dismissed by music theorists and composers as well. In part, this resistance stems from the rigidity by which Goodman believed scores identify compositions. According to the theory, only strictly notational [1] elements of a score are preserved with accuracy over successive reproductions of a score, and only performances that comply fully and exactly with the notational parts of a score can count as valid performances of a work. Experimental or graphic notations, which do not rely primarily on notation "scheme," cannot be trusted as preserving a work in a strict sense

Goodman's theory in fact sets such a high bar to work identification (the presumed purpose of scores) that on Goodman's account we likely never hear a genuine performance of any musical work or score, a fact not lost on many of his critics. In separate papers Paul Ziff and William Webster have convincingly argued that Goodman's theory of music notation failed to reflect the meaning and practical usage of scores, with Ziff additionally suggesting that Goodman overlooked the degree to which scores can only be accurately interpreted within the context of a particular performance-practice tradition [2, 3]. James Elkins has questioned whether the marks (the specific manifestations of notation on the printed page, as apposed to the interchangeable symbols) of music notation are truly indifferent, that is, whether the shape of the score elements, apart

[1] Goodman's use of the term "notation" refers to a strict usage of the term that differs substantially from the vernacular meaning. This distinction will be elaborated in section 1.2.

from their symbolic meaning, might have a significant effect on how musicians interpret a score [4]. Composer Jean-Charles Franois takes issue with Goodman's assumption that scores identify works (at least in the modern era) at all, preferring to consider the realization of a work alone to constitute that work [5]. Virginia Anderson notes that Goodman's rejection of graphic scores as non-notational leaves them in a kind of limbo, being far too score-like in their usage to be considered improvised compositions, while also apparently serving no work-preserving function, according to Goodman [6].

Despite its seeming shortcomings and paradoxes, Goodman's theory of music notation deserves reevaluation. Goodman brings attention to several often-overlooked questions: what kinds of information can be preserved in scores with fidelity? Is some score information more critical to the essence of a musical work than other information? And can studying notation give us insights into what musical parameters composers mean to preserve and which, if any, are contingent parameters? Additionally, Goodman's theory of notation allows music notation to be evaluated in the context of a general practice of notation in all disciplines, including, for example, scientific notation and data visualization. As composers increasingly make use of new tools for notation, including vector-based graphic design software and computer animation, it is important to develop philosophical paradigms for analyzing these works in a multidisciplinary graphical context.

While a purely notational account of contemporary scores may be implausible for reasons that will be elaborated in section 1.3, recent philosophical work on map semantics suggests that Goodman may have been right in his account of notations but mistaken about the fundamental representational modality of scores. In a discussion of maps, Goodman observed that road maps rely on a mix of analog and digital symbology. In a similar vein, John Kulvicki has observed that maps are "picture-language hybrids." It is striking that Goodman did not explicitly draw a parallel between the hybrid representation strategy of maps and that of scores. In emphasizing the notationality of scores, Goodman downplays the importance of other modalities of representation in scores, claiming that music notation "comes as near to meeting the theoretical requirements for notationality as might reasonably be expected of any traditional system in constant actual use, and that the excisions and revisions needed to correct any infractions are rather plain and local." This is not plausible, especially in the case of most contemporary scores where pictorial representations often significantly supplement or even replace traditional notation symbols.

In reframing music notation as a contingent feature of scores, a certain rigid conception of score-preservation and work-preservation must be sacrificed. However, if scores are in fact highly conventionalized maps, this account gains the ability to explain many special cases of contemporary score-making, use, and interpretation while revealing ways in which the syntax and semantics of maps function to represent a musical work through a sophisticated multimodal scheme.

1.2 Notation

For Goodman, notational systems are systems of symbols that represent things with a particular kind of fidelity. Notations section off and label certain parts of the universe, allowing information to be preserved without loss of accuracy due to subjective evaluation or imperfect reproduction. This distinction is the difference, for example, between recording a particular geometric angle in degrees or radians versus recording that same angle as a line drawing of an angle. Given consistent measuring equipment, an angle notated in radians can be reproduced with absolute fidelity, whereas an image may be degraded by subsequent reproductions.

An example of a strictly notational system is chess notation. At least one type of contemporary chess notation completely eliminates ambiguity from the recounting of a chess match. According to the "Figurine Algebraic Notation" (FAN) system of chess notation, each square on the board has a unique and discrete Cartesian coordinate. Furthermore, a unique pictogram represents each piece on the board (with the exception of pawns which are described by their rank and capture history). For example, moving the white queen two squares forward from her starting position is indicated in FAN by her symbol and destination coordinates, ♕d3.

Although we commonly refer to many kinds of symbolic depictions as notations, Goodman restricts this term to symbolic systems that fulfill strict criteria. The importance of defining a technical sense for the word notation, which may depart from the vernacular use of the word, is to explicate how and in what cases we can be confident that a symbol refers without ambiguity.

In order to be notational, the symbols that comprise a notational scheme must fulfill five criteria. Goodman's first two criteria relate to the syntax (or representational form) of symbols, while the remaining three criteria relate to the semantics of the symbols (or the content of these expressions). As Camp points out, this distinction between form and content is normally associated with linguistic expressions, but in the present case, it applies to any notational scheme and, as we will see later, is also relevant to the representational modality of maps. Goodman's five criteria for notational systems are as follows:

1. **The constituent symbols of a notational system must be disjoint (or "character indifferent").** In other words, marks that stand for equivalent symbols in a notational system must be capable of being exchanged without syntactic consequence. For example, in Figurine Algebraic chess notion, no symbol ever counts as an instance of more than one symbol in the system; e.g., there is no mark that stands in for both the symbol for the white queen and the symbol for the black queen. What matters is not that two characters be easily differentiated in practice—symbols may still be disjoint even if they are difficult to distinguish; such a notation would simply be an inconvenient notation, not an invalid one—rather it is the quality of belonging to only one class of

marks, (containing instances of a single symbol) that makes a notation disjoint or not.

2. **Symbols must be finitely differentiable, and such symbols are said to be "articulate."** That is, it must be theoretically possible to ascertain whether any two symbols in the scheme are disjoint. Goodman uses the example of a notation system composed of straight lines where lines are different symbols if and only if they differ in length to any degree. Since no test can ever guarantee that two lines do not differ in length by an unascertainably small margin, it can never be determined whether the lines are disjoint. Hence such a system is not articulate.

3. **The extension (or compliance-class) of a symbol must be unambiguous.** That is, the semantic referent of a symbol must be uniquely picked out by that symbol. In other words, regardless of when or in what context a symbol is used, the object represented by that symbol will always be consistent. For example, in Figurine Algebraic chess notation, the white queen is always referred to by the symbol ♕regardless of time or context.

4. **The semantics of the notational symbols must also be disjoint.** The set of objects to which a symbol refers may not overlap with the set of objects referred to by another symbol, e.g., redundancy within the field of reference is not permissible within a notation system.

5. **The compliants of a symbol within a notational system must be semantically finitely differentiated.** That is, it must be theoretically possible to determine that an object fails to comply with any given symbol in the notation.

Syntactically, symbols within a notation system may be composed of an indivisible unit ("atomic symbols") or composed of multiple atomic symbols ("compound symbols"). On Goodman's account, a musical score as a whole is a compound symbol that uniquely identifies a particular musical work. The purpose of a score is therefore to identify a particular musical performance with the musical composition of which it is an instance.

> "A score, whether or not ever used as a guide for a performance, has as a primary function the authoritative identification of a work from performance to performance" [6].

In order to uniquely identify a performance as an instance of a work, the score, as a notational symbol, must conform to Goodman's five criteria, which further entails that at least some relevant portion of the score must itself be based on notational symbols. For example, Goodman identifies pitch and rhythmic notation (the latter only in practice rather than in terms of its theoretical syntax) as being notational, at least as far as can be expected for a notational system in "traditional, actual use." On the other hand tempo indications (and presumably dynamics, glissandos,

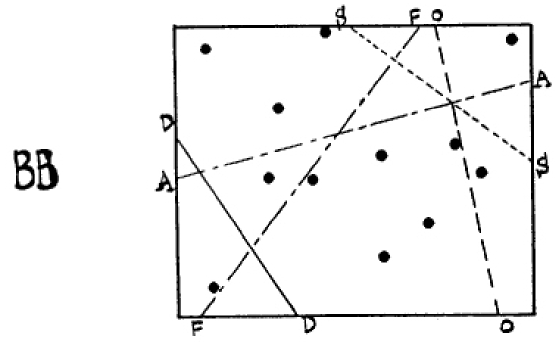


Figure 1. Without "stipulation of minimal significant units of angle and distance," p. 53 from John Cage's *Concert for Piano and Orchestra* from 1960 is not syntactically differentiated on Goodman's account. [2]

and much else), being syntactically dense, cannot be used to uniquely identify a score. Work identification cannot hang on any of these properties therefore, neither can any graphic score (Figure 1 in Goodman's own example) count as a score, since it contains no notational information.

1.3 Critical Response to Goodman

Although Goodman's economy of means is elegant, his conception of a score requires that, strictly speaking, we must reject the authenticity of any performance of a musical work that fails to conform to the minutiae of the relevantly notational elements of the notation. An imperfect performance of a work is not, in a strict sense, a performance of that work, because the score only represents performances that fall within its compliance class. Although we are free to speak casually of a performance being a performance of such and such a work, in a strict sense Goodman is adamant that a performance of a composition is only a realization of that composition if it is an exact realization of the score's notation.

> Since complete compliance with the score is the only requirement for a genuine instance of a work, the most miserable performance without actual mistakes does count as such an instance, while the most brilliant performance with a single wrong note does not. [...] If we allow the least deviation, all assurance of work-preservation and score preservation is lost; for by a series of one-note modifications, we can go all the way from Beethoven's Fifth Symphony to Three Blind Mice [6].

The strict sense in which a score identifies a work according to the score-as-symbol theory leads to some counterintuitive results. For example, since tempo marks are syntactically dense and hence not one of the relevant notational elements of a score on which work identification hinges, a

[2] Page 53 from John Cage's *Concert for Piano and Orchestra* is reproduced by permission of Edition Peters.

performance may still be an instance of a work even if it is played vastly faster or slower than the composer intended. The score for Beethoven's Ninth Symphony would theoretically still identify a performance of that work as an instance of the work even if the Ode to Joy were played over the course of an entire week or as a blur of nearly unrecognizable noise lasting only seconds, so long as the performers didn't actually miss or change the notes and rhythms of the work relative to each other.

It is also unclear exactly what it would mean for a musical work to be played according to the notational elements of the score. Although pitch and rhythm (in practice) are notational (at least in common-practice period notation) both of these parameters vary considerably depending on the performer, the circumstances of the performance, and the musical context in which the relevant passage occurs. For example, pitches in piano scores designate 12-TET tempered pitches, some of whose intervals are "out of tune" when compared to Pythagorean intervals. String players generally tune to Pythagorean intervals, except when they are playing with a keyboard instrument. Perhaps pitch notation is only notational in the context of a specific ensemble or for a specific player, but this too is challenged by the ubiquity of small pitch variations within even a short passage of music; in tonal music there is a tendency to raise the "leading tone" slightly; diminished tones are often played flat. Analyses of phonograph recordings of violin music found that violinists deviate from tempered pitches by 0.05 tones about 60% of the time and by 0.1 tones about 32% of the time [7].

The problem is not that notation requires absolute precision; semantic finite differentiation is sufficient to allow some tones to be identified as complying with no pitch in the notation (or at least this was the case before the ubiquity of microtonal music). Rather the problem for a notational conception of scores is that the symbolic representation of pitch seems to mean different things at different times, certainly between different instruments, but also even within a single phrase of music.

Imprecision in performed rhythms is pervasive and well-documented as well. Gabrielsson reports deviations of between 10-20% from the notated rhythm within two phrases of a Mozart piano sonata [8]. Various hypotheses are proposed for this variation ranging from expressivity to perceptual compensation or motorcontrol factors [9], but certainly such large rhythmic deviations bring the semantic disjointness of the notation into question.

Goodman gives us few hints as to how these problems might be reconciled. His project is fundamentally premised on providing a strict definition of score compliance, the criteria by which a performance may be judged to have been a performance of a specific musical work. Insofar as it is merely *impractical* to comply with all the notational information conveyed by a score, this is not a challenge to the theory. There is value, perhaps, in demonstrating the futility of ever actually performing a work of music according to a notation. (Some authors have in fact taken Goodman to have demonstrated that every musical performance is a kind of improvisation in a sense [10].)

Beyond various problems with the Godmanian notationality of music notation, the account doesn't seem to capture something important about the way musicians and composers interpret scores. A conception of score interpretation premised on producing precisely the correct referent of every notational symbol in a score seems stiff and contrived, what musicians refer to as "playing the notes" as apposed to performing music. Score interpretation has much more to do with context, finding how all the parts fit together or following a musical line or phrase. A change in tempo influences not only how we interpret the temporal symbols in the marked passage but also how we think about other passages of music, the purpose of that part of the music in relation to others, perhaps even how we think about the representational strategy of the score as a whole. Each symbol in a score affects the symbols around it and the work as a whole. This codependency of spatially and temporally representative abstract parts is a key feature of maps which will be discussed further in section 2.

2. A CARTOGRAPHIC THEORY OF SCORES

In arguing that scores are maps, I wish to make a claim about their syntactic and semantic strategy of representation rather than about the historical purpose or usage of maps. A "map" is therefore a broadly construed class of representations that may overlap in certain cases with what we might be more inclined to call graphs, infographics, diagrams, schematics, and charts. Camp argues that maps fall somewhere between pictorial and sentential modes of representation, and with some important qualifications, scores strike a similar balance.

Scores are maps that are isomorphic with the spatial and temporal structures of the musical works they represent, while other graphical features may be purely contingent or incidental. This highlights an interesting property of maps: they need only be isomorphic with regard to a subset of the properties of the space they represent. A true subway map must be isomorphic to the order and correlation of subway stations, but almost every other property of the landscape can be omitted or abbreviated symbolically. What aspects of the world are represented and what aspects are omitted or stylized would seem to have a great deal to do with the power of maps to expand and clarify our understanding of specific spatial relationships. Camp highlights that the choice of features depicted is connected to the practical function of a map.

> "[...] typically this spatial isomorphism itself only captures functionally salient features of the represented domain: for a road map, say, only streets and buildings and not trees and benches.[1]

Unlike road maps which represent a certain geography, albeit from a "disengaged, 'God's eye' perspective" [1], scores represent an array of highly structured acoustic morphologies and performative actions through two-dimensional, visual conventions. In essence, scores translate a specific subset of acoustic and temporal features of their referents

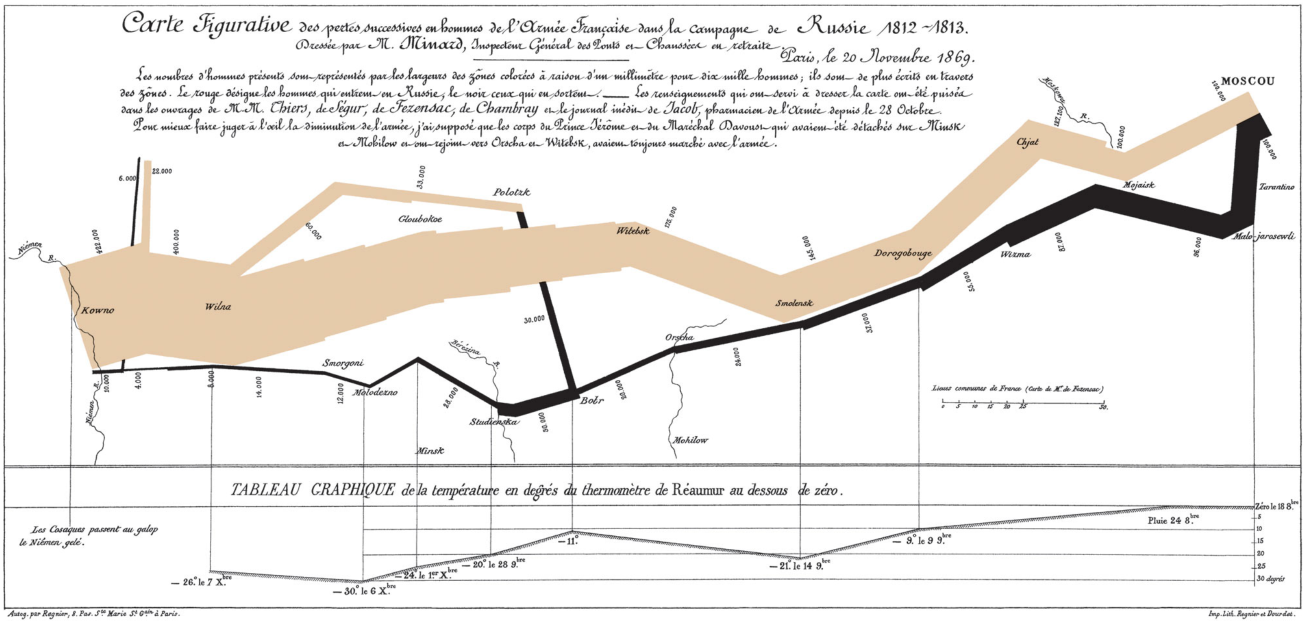


Figure 2. Charles Minard's 1869 graphic representing Napoleon's disastrous Russian campaign and an excellent example of a cartographic representation of spatial and temporal events (months, shown with Roman numerals, are correlated to temperature and spaced according to the distance between landmarks).

to a visual representation. Because of this, what constitutes isomorphism is far more conventionalized in scores than in most maps, depending on cognitive metaphors to translate back and forth between spatial and temporal domains rather than simple visual similarity. Within these conventions, isomorphism is preserved however. "Higher" pitches appear visually higher on musical staves; rhythms are ordered as they occur in time from left to right; in percussion music, instruments are grouped as they appear before the performer, with each instrument in a collection assigned to a line on a special staff (or a syntactically disjoint symbol, e.g. a notehead of a certain shape).

As in many maps, scores can use sentential representation, and through the map-like structure of the score, these expressions gain the ability to refer to specific temporally and spatially locatable features. Performance directions can be far more conceptual than can be easily expressed either through pictures or diagrams, with Pierre Boulez, for example, calling on the performer of his Second Sonata to play in an "exasperated" or "strident" manner or, later, to "pulverize the sound." Through the map-like scheme, these abstract invocations are applied only to certain sections of a work, thereby increasing the expressive power of language beyond the contents of the sentential expression.

Scores are not themselves sentential in structure however, since they lack the extremely hierarchical and abstract structure of language. Disregarding aesthetic or stylistic concerns, the discrete parts of a score can generally be rearranged with a great deal of freedom, and musicians even refer to these parts in spatial terms. A musical line can be "inverted"; melodic lines are said to be "close" or to "cross" or contain too many "leaps"; harmonies are said to "revolve" around a "harmonic center."

2.1 Representing Objects and Events

The most striking characteristic of scores—that they represent objects and events with regard to time—is an uncommon but not wholly neglected, feature of maps. Gail Langran and Irina Vasiliev have documented cartographic practices of depicting time, with Vasiliev dating the earliest examples back to at least the 18th century [11, 12]. We are most familiar with temporal map-like depiction from animated maps such as weather maps, traffic maps, or animated subway maps (although it is debatable in what circumstances animated maps represent time or whether they actually *depict* time through a real-time change in the image). [3] Among printed maps, excepting scores, representations of temporal processes are generally only achieved crudely. Maps of historical battles often depict the movements of military units with arrows. Maps of population growth and migration show the expansion of species or living organisms over geography in a very general way.

A notable early exception to this is Charles Joseph Minard's illustration of Napoleon's 1812 invasion of Russia (Figure 2). Unusually for a map, this illustration correlates time with multiple other domains of information, showing landmark dates during the disastrous fall and winter retreat correlated to both temperature and geographical movement of the army. Something similar usually occurs in scores, where musical time flows differently according to circumstances, being modified by tempo indications or rubato, for example, while at the same time, temporal events are tightly bound to a vast array of spatial and performative information.

In a brief survey of the philosophical literature on the spatio-temporal analogy, Robert Casati and Achille Varzi

[3] Further research is needed to establish a theory of animated maps that might inform the design of animated scores.

note two schools of thought [13]. Bertrand Russell [14], Alfred Whitehead [15], and Willard Quine [16] generally held that physical objects and temporal events are highly analogous, while the disanalogies stance is tokened by David Wiggins [17] who objected that the boundaries of spatial objects may be explored while this is not obviously true of temporal objects. If true maps are admitted as a means by which a continuant is explored, then temporal maps, and particularly scores, offer an interesting challenge to Wiggins' contention. Perhaps scores do not offer insight into specific events, but they do allow inferences about planned or hypothetical events such as the performance of a particular piece of music. Casati and Varzi outline a "formal map," which is to informal maps as formal logical languages are to vernacular language. A temporal referent would not appear to be any barrier to the creation of a "formal score" with an analogous form.

As with a cross that represents a church on a road map, expressive or technical directions in scores are represented by arbitrary designators (symbols or words) that stand in complex relationship to numerous other features of a musical work. These designators fall before or after other features; they apply to specific instrumental parts; and they last for finite durations, dividing up the temporal space of a work as a map is divided between "land" and "water." As with symbols on a road map, we may even be unaware of the meaning of a symbol, but as long as we are familiar with the isomorphic, spatial strategy employed by the map, we can make valid statements about that symbol in relation to others in its vicinity.

Interestingly, Gennady Andrienko, et al., note a temporal corollary for "Tobler's first law of geography" ("everything is related to everything else, but near things are more related than distant things" [18]). Referred to in its spatial manifestation as "autocorrelation," the principle that closely spaced spatial features are dependant on one another is seen in the temporal domain as well in the connection between past, present, and future. Temporal features run forwards and backwards through time, with experience of the past and anticipation of the future both informing the present [19]. This principle certainly holds for scores, where for example "courtesy accidentals" are used to confirm the cancellation of a change in pitch that occurred earlier in a passage of music.

2.2 Conventions of Representation in Maps

Goodman denied that pictures depicted through resemblance with their subject. Rather, Goodman believed that depiction was almost entirely a matter of artistic convention. Without wishing to take a position on this question here, the same cannot hold true for maps. Maps may be conventionalized to a very high degree in their non-semantically relevant properties. A 2012 New York City subway map redesigned by Max Roberts uses only sections of concentric circles, abstracting away nearly all information about absolute distance or geographic movement vector [20]. The interaction between the syntactically relevant representative components of a map must still stand in an isomorphic relationship with the depicted properties of the landscape however, or else the map is inaccurate. In the case of a subway map, the subway stations must occur in the correct order, although a wide variety of symbols and labels may stand in for the stations and the subway lines themselves. Concerning the symbolic constituents of a map, Camp has observed that they too exhibit some limits to their abstraction:

> "[A]lthough maps employ discrete syntactic constituents with a significantly conventionalized semantics, there's still a significant interaction between their formal properties and mode of combination and what they represent. Nonetheless, the only strong constraint on the icons employed by cartographic systems, and on their potential semantic values, is that the icons' own physical features can't conflict with the principle of spatial isomorphism. Thus, one can't represent a street with a circle, not because it would be too arbitrary, but because this would make it impossible to place the icon in a spatial configuration that reflects the spatial structure of the represented content: for instance, one couldn't depict two streets as parallel, or as intersecting" [1].

Maps differ from pictures in that they abstract away much of the detail of pictures, increasing comprehension by replacing complex depiction with symbolic representations while preserving certain relevant spatial relationships between these constituents. Cities and towns are replaced with pictograms or labels. Roads and highways are lines of different colors. Colored patches represent areas of water or forest.

Different types of maps abstract different features and range in their level of detail from, as Camp points out, Google Maps renderings that allow for satellite and street-view images to be overlaid over roadways (at the less abstract end of the spectrum) to subway maps and seating charts (some of the most abstract maps in common usage). By depicting certain properties as highly isomorphic while others details are omitted or stylized, map designers affirm the importance of certain kinds of information and relationships while downplaying other details. It is vital to the success of a map that it be isomorphic in the properties most vital to a map's intended usage. A nautical map must reflect the depths of oceans and waterways, while a road map need only show the location of water. Depicting the depth of water on a road map would only serve to distract from the map's intended purpose.

Similarly, composers adopt a position on what criteria are vital to the essence of a musical work when they prioritize certain types of representation in their scores. These choices, which I refer to as "work-preserving criteria" and "score-preserving criteria," suggest a different model of work preservation from that advocated by Goodman. Instead of basing work preservation on a score functioning as a compound notation for a work, works preserve only relevant isomorphic features, and these are used to navigate a temporal and acoustic space suggested by the composer.

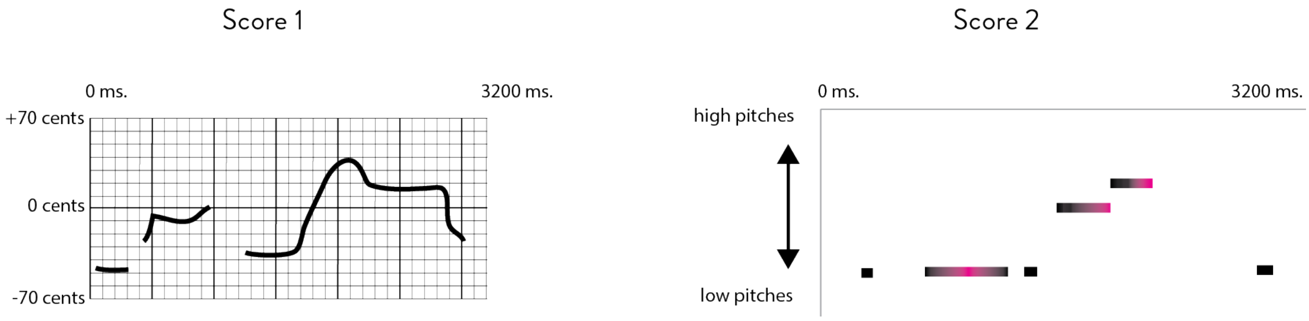


Figure 3. Two hypothetical scores committed to two different scales and exhibiting very different score and work-preserving criteria.

3. SCORE DESIGN CONSIDERATIONS

Although the philosophical underpinnings of a cartographic theory of scores require a more thorough exposition elsewhere, my intention here is to consider a practical theory of cartography that can yield insights into notational practices. In what follows I will discuss two aspects of map idioms: scale and coverage.

3.1 Scale and Preservation

Regarding score preservation in graphic systems such as Figure 1, Goodman worries that "however small the inaccuracy of reproduction, a chain of successive reproductions can result in departing to any degree from the original." Finitely differentiated symbols do preserve information better across successive reproductions; while as the symbols that comprise the notation are identifiable, their compliance class is fully intact.

Slight imperfections of reproduction have less semantic significance on a score-as-map conception. The accuracy of a map is only valued according to how it is used. A key to informational density, the scale of a map is intended to give some indication of what kinds of uses a map might be good for. If we wish to know the travel time required to drive from Bremen to Stuttgart, we may be happy with a map that represents distance in kilometers. On the other hand, if we must know the location of the gas line entering a house, only a map or diagram representing distances in inches or centimeters will suffice. The degree of inaccuracy we are willing to accept in a map depends on what we want to do with the map. Similarly, different musical works accept different levels of inaccuracy, and according to the conventions of the style, we may or may not be inclined to accept a particular performance as a genuine instance of a work depending on the degree to which the performance departs from the score.

The representational scheme chosen by the composer always necessarily prioritizes certain kinds of accuracy of reproduction while deemphasizing other less salient syntactic components. Furthermore the choice of a particular scheme implies that certain syntactic components will receive more consideration in making judgments of work preservation than others.

For example, Figure 3 represents two hypothetical scores. "Score 1" leverages Cartesian graph notation and allows for nuances at least down to tens of cents. By choosing to represent this kind of detail, the composer implicitly takes a position on the "scale factor" for the score, which in turn has implications for the score-preservation criteria for the work. The score is not fully notational by Goodman's standards. For that we would need a syntax for the contour line including notation for angles, path lengths, etc. However, we can infer that an existential threat to the score would be one that prevents us from interpreting the contour paths with accuracy on the order of tens of cents. We can also make map-like intuitions that will constrain the inaccuracy of the contour paths within the limits of the scale factor. For example, we can note that the first contour in the work (beginning between 0 and 100 milliseconds) is in the third space up from the bottom of the graph and is just touching the third line up from the bottom.

It is true that over successive reproductions of the score, the exact path traced by the contour line may be affected by successive inaccuracies in the reproduction process (as in Goodman's score-preservation challenge to Cage, Figure 1). However, by not defining the contour line's path more strictly, we should understand that the composer is implicitly assenting to the proposition that score preservation still holds *so long as the contour line does not depart too far from the constraints of the scale.* In other words, a change of 20 cents in contour line morphology would destroy score preservation. A change of 3 cents (hardly visible on the score) does not threaten score preservation, and any change much more than this will be rapidly detected in relation to the graph. The graph lines, like cartographic symbols for longitude and latitude, are notational and therefore limit the degree to which the analog parts of the score could conceivably deviate from the manuscript.

In Figure 3, "Score 2," the composer has implied a different scale and hence very different score-preservation criteria. Here our only indication of pitch is a range between high and low. By constraining pitch only loosely, the composer implicitly assents to the proposition that precise pitch is not a factor that affects score preservation. Rather, the map-like syntax implies that the ordering of pitches is mandatory, and changing the order of high and low pitches would pose an existential threat to score preservation. Similarly, by not providing a graph by which to compare note lengths in milliseconds, the composer is assenting to a scale factor that requires performers to follow

Figure 4. Detail of freeway map of Los Angeles redesigned by Peter Dunn. [4]

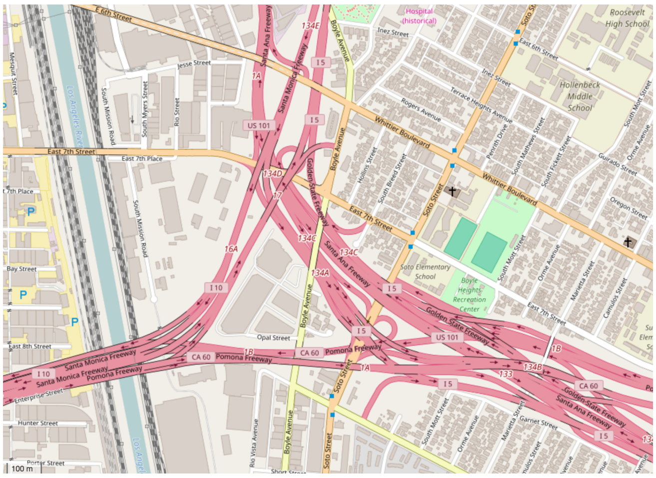


Figure 5. A typical color scheme for road maps is based on a highly isomorphic representation of locatable features but may be disorienting for wayfinding [5]

only very approximate note durations. In this case, playing a notated short note for a longer duration than a notated long note would pose an existential threat to score preservation, but minor inconsistencies in note duration are tacitly permitted.

3.2 Coverage and Degrees of Freedom

The notation in Figure 3, "Score 2," differs from "Score 1" in another important regard: the use of a third color (in fact a color gradient) allows this notation to refer to an additional "degree of freedom," perhaps dynamics or a timbrel effect.

Kulvicki refers to degrees of freedom as features of a map that, once introduced, have communicative significance across the relevant portion of the map [21]. In Kulvicki's example, for instance, a simple map may be silent as to whether a green "land" area of a map is flat or mountainous, but once a squiggly line is introduced to represent hilly terrain, then unblemished green has an additional meaning within the degree of freedom that encompasses the binary "hilly" versus "relatively flat" terrain. Each degree of freedom a map represents commits that map to representing the null value for that degree of freedom wherever a space is left unmarked; if a map commits to representing towns symbolically, the absence of a "town" symbol commits the map to an absence of a town at that location.

Like maps, scores are agnostic with regard to all degrees of freedom save the ones introduced into the score by the composer. Scores that represent only pitch information say nothing at all about rhythms, or rather they imply that durational information must be improvised up by the performer, either by relying on conventions or through other, perhaps sentential, instructions. (This is the case for some of John Cage's freer "Number Pieces," such as *Four*^{3}, for example.) Once a notation for rhythm is introduced into a score, however, passages without rhythmic notation imply a special significance; The scale of a score, the level of detail it commits to representing (i.e., its score-preservation criteria) are degrees of freedom, because they represent the detail that a composer has represented as important for the particular work of which the score is a map. In contemporary notation where performance practice fills in very little for a performer, a score that is agnostic as to note durations implies that the composer is explicitly declaring durational plasticity to be a score-preserving feature of the work. To represent specificity with regard to some features in conjunction with agnosticism about others is to make a statement about what features of a work are valuable to the composer of that work.

Kulvicki also defines an "incompatibility constraint": for well-formed maps "incompatible locatable features represent incompatible qualities." For example, yellow lines can be used for interstate highways and purple lines can be used for county highways. The incompatibility of these colors (no road can be both yellow and purple) is in accordance with the incompatibility of their referents no road can (ordinarily) be both an interstate highway and a county highway. If however, green is then introduced to indicate a toll road, then the incompatibility constraint will possibly be violated, since a county highway may be a toll road as well.

Although colors don't admit much granularity due to the impracticality of perceiving different similar shades of a single color, within degrees of freedom represented through other symbolic means, complex information can be filled out extensively without risk so long as the locatable features of the referents are all incompatible.

> Mountain marks pair with smooth texture as mutually incompatible, but syntactically significant, aspects of a map. Once mountain texture is on the menu, it is easy to add more textures for different kinds of land: alps, pied-

[4] Detail of Greater Los Angeles Freeway System Map reproduced by permission of Peter Dunn. Image from http://www.stonebrowndesign.com/

[5] OpenStreetMaps cartography is licensed under Creative Commons Attribution-ShareAlike 2.0. Image from http://www.openstreetmap.org/

> mont, hills, bumps, etc. Each of those textures is incompatible with the others, and what each represents is incompatible with what the others represent. Untextured, smooth areas are the zero value along this degree of freedom. Being smooth carries representational weight just as the marks do [21].

In practice, when symbolic schemes violate their incompatibility constraints, redundant representational strategies can sometimes prevent a critical failure of coverage from occurring. In Peter Dunn's beautiful redesign of the LA freeways system map (Figure 4), the Santa Monica Freeway (blue) briefly passes through and co-designates a short stretch of the Golden State Freeway (yellow). Dunn solves the incompatibility of the color designators by replacing the solid line of the Santa Monica Freeway with a dotted line for the portion of the two highways in which they overlap.

Similar ambiguity is frequently encountered in scores. "Hairpins," which indicate a change in volume, have an ambiguous meaning when they pass under rests (Figure 6). Although several incompatibilities seem to be at work here, the most promising way to explain the problem is that hairpins refer to an interpolation of sound intensity over the duration of one or more sounds. A hairpin under a rest may be syntactically sound but semantically flawed. Certain composers, notably Brian Ferneyhough, have adopted a dotted notation that clarifies this ambiguity. As in Dunn's map, a redundant symbol (the dotted hairpin) is incorporated into the map only in case the primary symbol encounters incompatible features.

Dunn's use of color gradients to symbolize transition points between symbols within a single degree of freedom (represented by colored solid lines—highways) is particularly notable. Color and abstraction of vectors is a key to a design that illuminates opaque aspects of a more traditional, highly isomorphic OpenStreetMap visualisation (Figure 5). In Dunn's map a gradient indicates an exit or on-ramp whereas a mitered join indicates an overpass or underpass. This layering of representational strategies through color, shape, and spatial organisation is a key to the ability of maps to represent numerous dimensions of information in an abstract gestalt unit.

Whereas traditional notational strategies for "extended techniques" (atypical means of producing sound on an instrument) conventionally rely on introducing different symbols along different degrees of freedom for each new specified extended technique, a more idiomatic cartographic representation would represent incompatible extended techniques as symbols featuring incompatibility constraints.

For example, a traditional symbolic notation for *col legno tratto* (bowing a string instrument with the wood of the bow) potentially yields an impossible map when it is erroneously layered with a symbolic notation for scratch tone (playing with extreme pressure of the bow hair) (Figure 7). Since playing heavily with the bow hair is largely incompatible with bowing with the wood of the bow, these two

[7] Score excerpt from *this will be changed and made solid II* used by permission of the composer.

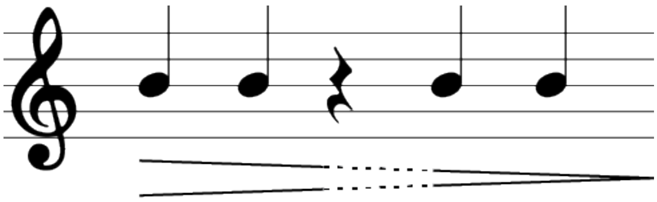


Figure 6. Hairpin with dotted-line notation to resolve ambiguity created by passing under rests.

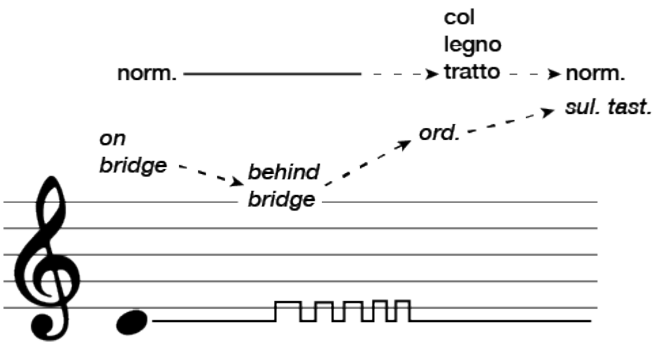


Figure 7. Traditional notation of extended techniques often exhibits poorly formed incompatibility constraints that allow for impossible layering effects. Too many types of representation, including words, symbols, and spatial distribution, are used simultaneously leading to difficulties in viewing the notation as a gestalt.

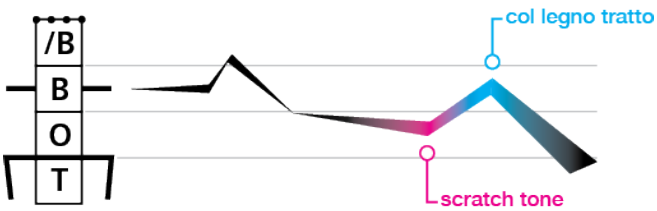


Figure 8. A hypothetical notation with well-formed incompatibility constraints and a more isomorphic graphic approach.

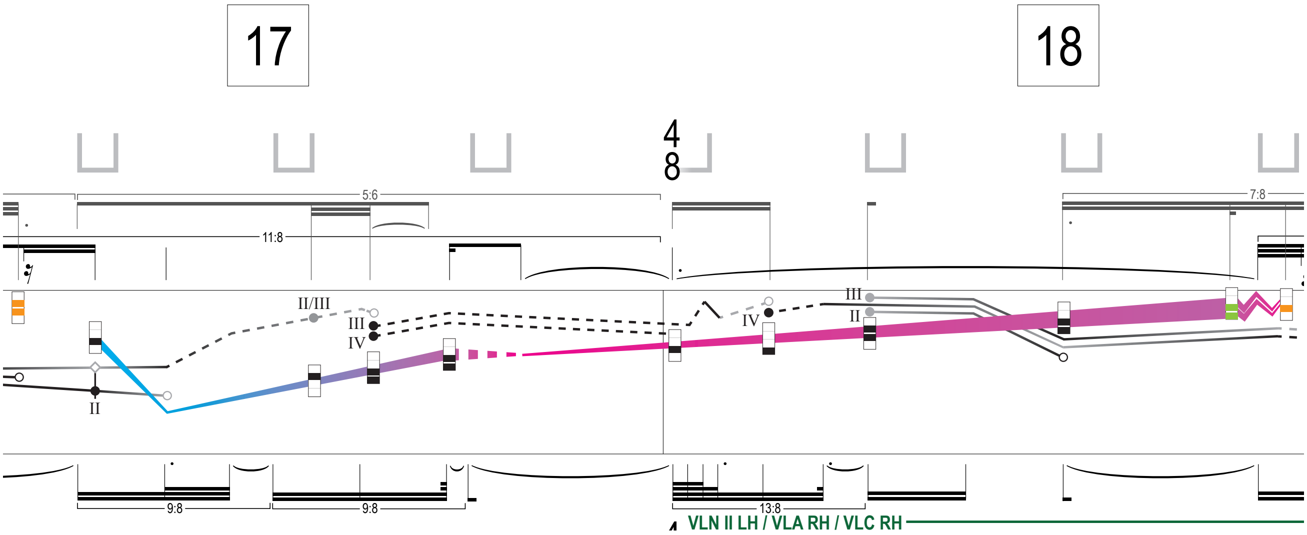


Figure 9. An excerpt from *this will be changed and made solid II* for string quartet by James Bean. Color represents bow placement; colored-line thickness represents bow pressure; colored-lined vector represents bow movements; vertical, striated rectangles represent the string to be bowed, while the height of these "string indicator boxes" represents the maximum bow pressure; black or dashed vectors and filled or open circles/diamonds represent finger movements along the strings with the indicated amounts of pressure; rhythms above the staff indicate temporal placement of right-hand information while rhythms beneath the staff are for the left-hand. [7]

techniques should be represented through an "incompatibility constraint" in the score. In Figure 7 the extensive layering of different symbolic strata is difficult to perceive as a gestalt, and it is nearly impossible to tell how and when the scratch tone becomes *col legno tratto* or normal tone. A separate degree of freedom has been introduced for pitch, bow placement, and two different degrees of freedom are used to notate bow timbre. Yet another degree would be required to show dynamics, although this too might be incompatible with the overpressure which is difficult to execute softly.

Figure 8 shows the notation of a similar passage to that shown in Figure 7. However, incompatibility-constrained symbols (the colors cyan and magenta) are used to show *col legno tratto* and scratch tone respectively. As in Dunn's map, the use of semantic incompatibility to show syntactically incompatible features entirely clarifies the smooth transition between the timbrel techniques. Additionally, to aid in the formation of a gestalt representation, line thickness is here used to show bow pressure and vertical spatial distribution (isomorphic to the body of a string instrument) is used to notate bow placement. [6]

An example of interesting and well-designed spatial-temporal scheme is found in *this will be changed and made solid II* for string quartet by James Bean (Figure 8). Written in 2012 and relying on extensive unconventional graphics created in the Adobe Illustrator software environment, the score is an interesting and complex type of tablature. By notating only the movements and actions of the right and left hands, the score remains consistent in its field of reference. There is no need to switch between perceptual descriptors (dynamics, expressive bowing, etc.) and physical actions (the absolute pressure of the bow on the string, its placement on the instrument, the movement of the left hand fingers, etc.). At the same time, by placing all tactile information in the center of the staff and reserving the extremities for rhythmic notation, the eye is better able to track a melodic gesture as a single multidimensional contour.

Bean's score takes the form of several maps layered on top of one another within the same representational space. For vectors relating to the movement of the left hand, the top of the staff is to be considered the bridge while the bottom of the staff symbolizes the nut. For vectors relating to the movement of the right hand, the top of the staff represents the frog of the bow while the bottom of the staff is the tip. Further research should consider what effect this multilayered scheme might have on map perception, coverage, and incompatibility of degrees of freedom and whether there are examples of multilayered representations in other fields.

4. CONCLUSIONS

The practice of score-making in the 20th and 21st centuries has become so varied and complex that it is impractical for a single theory of representation to encompass all cases. Certainly there are scores, such as Karlheinz Stockhausen's *Aus den sieben Tagen,* that operate on an entirely sentential basis. It is harder to think of scores that very closely approximate Goodmanian notation, but piano rolls—the long perforated paper scrolls that are the con-

[6] Although inspired by Helmut Lachenmann's "bridge clef," the spatial depiction of the string instrument is here modified to be visually discrete in its vertical layout. The use of Frutiger typeface, a high-visibility font designed for signage in Charles de Gaulle Airport, is also an innovation borrowed from "wayfinding" design. Space does not permit a discussion of the interesting parallels between wayfinding and cartographic modes of representation. The letter abbreviations are inspired by a system of bow placement indications used by Timothy McCormack.

trol interface for a mechanical player piano—might count if they are in fact a kind of score.

In Cage's 1969 book *Notations*, Jean-Charles Franois finds a hint of the doubts Goodman sought to answer in *Languages of Art*.

> "As soon as there is a necessity to demonstrate unequivocally that there is something to show, one has to persuade oneself that there is something to be shown. Here we find an infinite nostalgia for an ancient world in which the question of representation would never have been asked or considered in the first place."

Notations, a collection of pages from scores in a wide variety of graphic styles, signaled a change in the traditionally held view of scores [22]. No longer were music scores to be regarded as the "crystal goblet," as Beatrice Warde famously said of good type in her 1955 essay ("The book typographer has the job of erecting a window between the reader inside the room and that landscape which is the author's words." [23]). Instead, *Notations* is premised on a very modern conception of the score as a multimedia, multimodel object whose relation to the musical work is complex, often abstract or indirect, and highly conventionalized but nevertheless capable of expressing complex relations between objects and events in space and time that would not easily be conveyed in sentential form.

The increasing adoption of new software paradigms for notation combined with highly specific, systematised, or graphic notations developed by composers such as Timothy McCormack, Aaron Cassidy, and Cat Hope suggests the importance of developing a philosophical approach that can better analyze multiple modes of representation as functioning simultaneously within a temporally and spatially isomorphic representation. A cartographic theory of scores gets us a little bit closer to untangling that complexity.

5. REFERENCES

[1] E. Camp, "Thinking with maps," *Philosophical Perspectives*, vol. 21, no. 1, pp. 145–182, 2007.

[2] P. Ziff, "Goodman's languages of arts," *The Philosophical Review*, vol. 80, no. 4, pp. 509–515, 1971.

[3] W. Webster, "Music is not a 'notational system'," *The Journal of Aesthetics and Art Criticism*, vol. 29, no. 4, pp. 489–497, 1971.

[4] J. Elkins, *The domain of images*. Hackett Publishing Company, Inc., 1999.

[5] J.-C. Franois, "Writing without representation, and unreadable notation," *Perspectives of New Music*, vol. 30, no. 1, pp. 6–20, 1992.

[6] N. Goodman, *Languages of Art*. Hackett Publishing Company, Inc., 1976.

[7] A. Small, "An objective analysis of artistic violin performance," *University of Iowa Studies in the Psychology of Music*, vol. 4, pp. 172–231, 1936.

[8] D. Deutsch, *The Psychology of Music (Cognition and Perception)*. Academic Press, 1998.

[9] A. Penel and C. Drake, "Timing variations in music performance: Musical communication, perceptual compensation, and/or motor control?" *Perception Psychophysics*, vol. 66, no. 4, pp. 545–562, 2004.

[10] D. Ake, *Jazz Cultures*. University of California Press, 2002.

[11] G. Langran, *Time in Geographic Information Systems*. Taylor and Francis, 1992.

[12] I. Vasiliev, "Mapping time: An analysis of the cartographic problem of representing spatiotemporal information," Ph.D. dissertation, Syracuse University, 1996.

[13] R. Casati and A. Varzi, *Parts and Places*. MIT Press, 1999.

[14] B. Russell, *The Analysis of Matter*. Spokesman Books, 1927.

[15] A. Whitehead, *Process and Reality*. Free Press, 2010.

[16] W. Quine, "Identity, ostension, and hypostasis," *The Journal of Philosophy*, vol. 47, pp. 621–633, 1950.

[17] D. Wiggins, *Sameness and Substance*. Harvard University Press, 1980.

[18] W. Tobler, "A computer movie simulating urban growth in the detroit region," *Economic Geography*, vol. 46, pp. 234–240, 1970.

[19] G. Andrienko, N. Andrienko, U. Demsar, D. Dransch, J. Dykes, S. I. Fabrikant, M. Jern, M.-J. Kraak, H. Schumann, and C. Tominski, "Space, time and visual analytics," *Journal International Journal of Geographical Information Science*, vol. 24, no. 10, pp. 1577–1600, 2010.

[20] M. Roberts. (2013) The story of circles maps. [Online]. Available: http://www.tubemapcentral.com/circles/NY_Circles_a.jpg

[21] J. Kulvicki, "Maps, pictures, and predication," *Ergo*, vol. 2, no. 20161109, pp. 149–174, 2015.

[22] J. Cage, *Notations*. Something Else Press, 1969.

[23] B. Warde, "The crystal goblet," in *The crystal goblet : sixteen essays on typography*, H. Jacob, Ed. World Pub. Co, 2056, ch. 1, pp. 11–17.

HOW CAN MUSIC VISUALISATION TECHNIQUES REVEAL DIFFERENT PERSPECTIVES ON MUSICAL STRUCTURE?

Samuel J. Hunt
University of The West of England
Department of Computer Science
& Creative Technologies.
Samuel.hunt@uwe.ac.uk

Tom Mitchell
University of The West of England
Department of Computer Science
& Creative Technologies.
Tom.mitchell@uwe.ac.uk

Chris Nash
University of The West of England
Department of Computer Science
& Creative Technologies.
Chris.nash@uwe.ac.uk

ABSTRACT

Standard western notation supports the understanding and performance of music, but has limited provisions for revealing overall musical characteristics and structure. This paper presents several visualisers for highlighting and providing insights into musical structures, including rhythm, pitch, and interval transitions, also noting how these elements modulate over time. The visualisations are presented in the context of Shneidermans Visual Information-Seeking Mantra, and terminology from the Cognitive Dimensions of Music Notations usability framework. Such techniques are designed to make understanding musical structure quicker, easier, less error prone, and take better advantage of the intrinsic pattern recognition abilities of humans.

1. INTRODUCTION

Standard western notation serves as a strict, formal set of instructions for the performance of composed music. However, it omits explicit representation of a rich amount of hidden data that exists between individual notes, and the location of the notes within an overarching musical structure. One way to understand this structure is to analyse the music: either manually, requiring an experienced musicologist; or via computer, resulting in several multi-dimensional data fields, which may be difficult to represent and comprehend. Representing this data visually utilises the brains pattern detection abilities, supporting easier and faster comprehension of material to enable insight and speculation that can inform further formal analysis.

Visualisation presents non-visual data in a visual format, usually as 2D/3D images or video. Shneiderman [1] introduces a framework for guiding the design of information visualisation systems, known as the Visual Information-Seeking Mantra (VISM). The framework consists of seven tasks for presenting information in a visual form to a user (Table 1). Craft and Cairns [2] elaborate on this by stating the VISM serves as inspiration and guidelines for practitioners designing visual information systems. /par

Task	Description
Overview	Gain an overview of the data.
Zoom	Zoom in on items of interest.
Filter	Filter out uninteresting items.
Details-on-Demands	Selected an item or group and get details when needed.
Relate	View relationships between items.
History	Keep a history of actions to support undo, replay, and progressive refinement.
Extract	Allow extraction of sub-collections and of the query parameters.

Table 1. The 7 tasks of the VISM.

Shneiderman emphasises that humans have remarkable perceptual abilities, allowing them to easily detect changes of and patterns in size, colour, shape, movement or texture in visual media. Such advanced and robust feature extraction capabilities are considerably more difficult to encode as automated analysis using computer systems.

In a musical context, visualisers also enable rapid, automated methods for visualising not only a single piece of music, but an entire corpus - allowing understanding and comparisons of musical material at a higher and more generalised level to that of manual score analysis.

The level meter which features in the majority of consumer audio products, represents a ubiquitous visualisation method, whereby the current sound level is visualised using vertical bars, and for the majority of situations a more useful presentation than a display of audio sample values (amplitudes). Digital audio is stored as a series of numbers, a sequence of amplitude measurements with respect to time. Sonograms convert this information to visualise the distribution of frequency content. An example of this is illustrated in Figure 1, whereby the musical score has been synthesized using piano samples on a computer and analysed with a sonogram.

This paper focuses on visualizing scores at the note-level (e.g. MIDI), avoiding the many difficulties of audio feature extraction. Sequenced music, encoded as MIDI, by contrast allows for rapid and reproducible analysis [3]. The aim of the paper is to present novel techniques that support the analysis of music.

The remainder of the paper is broken down as follows.

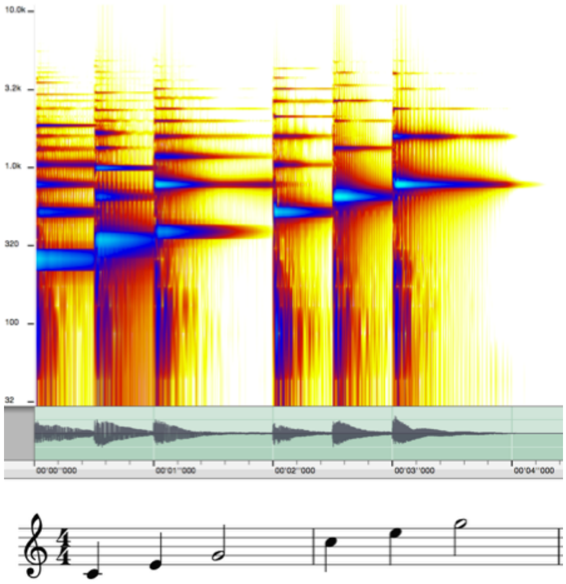


Figure 1. Sonogram plot of the score.

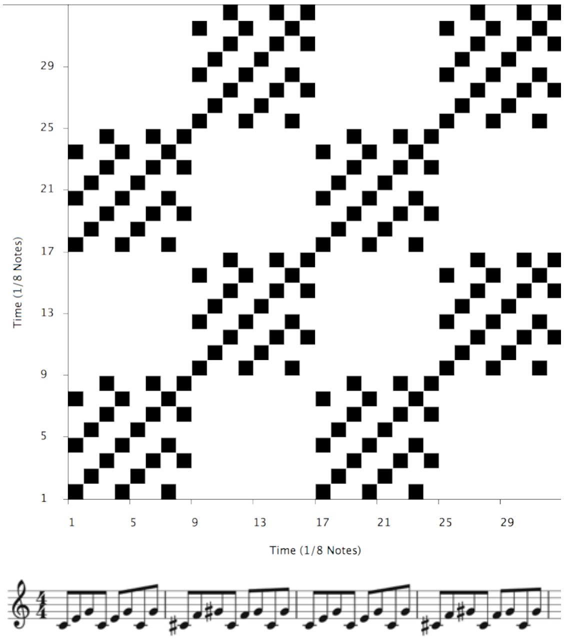


Figure 2. Self-Similarity visualisation of the score.

Section 2 presents relevant prior work and theory, followed by a brief discussion in Section 3 of the software system developed to support this research. Section 4 reviews visualisation techniques for pitch, contours, intervals and key, followed by Section 5 looking at rhythmic elements. Section 6 discusses visualisation techniques that integrate both pitch and rhythmic elements. The final section considers future work for the area and proposes evaluation techniques.

2. RELATED WORK

Prior work in music visualisation can be broadly categorised into two groups: those exploring sampled audio data and those exploring sequenced music data (scores and MIDI). Soriano et al [3] present methods for browsing an audio-based music collection, using graphical metaphors designed to convey the underlying song structure. This analysis is performed via feature extraction from MIDI files, enabling easy identification of simple and meaningful musical structure, such as pitch and rhythm.

Foote [4] and Wolkowicz & Brooks [5] both used self-similarity matrix visualisations to reveal similarity in music. This visualisation approach relies on the measurement of pitch content at quantised time intervals, and plotting this against all other intervals. Figure 2 shows a self-similarity matrix visualisation, whereby the music proceeds through time from the bottom left to the top right, with regions of similar patterns appearing as clusters of squares. Both axes represent the same input vector. The music example uses a repeating motif of one bar, with a modulation at bars 2 and 4.

Bergstrom [6] presents several visualisers that convey information about interval quality, chord quality, and the chord progressions in a piece of music, helping users to comprehend the underlying structure of music. Feedback from engagement with the system revealed users who having quickly understood the basics, wanted to compare music from multiple genres and composers. Holland [7] presents a similar system (Harmony Space) to allow beginners to interact with harmony using a visual grid.

Jeong and Nam [8] discuss a system that visualises audio streams, to show audio features such as, volume, onset density, and dissonance. The authors also state that as music is an auditory art, visual representations can contain information that cannot be transferred or perceived accurately with sound. Herremans and Chew [9] use visualisation to highlight tonal tension in music, creating an explicit representation of something that is not easily quantifiable, presenting graphics alongside the scored elements.

Established analytical frameworks for music, such as the Generative Theory of Tonal Music (GTTM) [10] and Schenkerian analysis [11], also present ways to annotate music and reveal structure. The GTTM proposes a series of preference rules for determining the different musical structures that underlie the perception of western music. Schenkerian analysis is an established musical analysis technique that aims to explicitly reveal hidden dependencies and structures implicit in the music. This analysis primarily aids score reading by marking it with elements of musical structure. Both of these theories have been mechanised in software [12][13].

Nash [14] presents research that adapts the Cognitive Dimensions of Music Notations framework (CDMN) [15], for use in designing and analyzing music notations and user interfaces for digital and traditional music practice and study. This paper utilises the framework as a vocabulary for comparing visualised music content and metadata against western notation and other forms of visualisation. However not all of the 16 core dimensions originally specified are of relevance here. A list of the terms and their definitions relevant in this research are listed in Table 2.

Using the self-similarity visualisation in Figure 2 as an

Dimension	Description
Visibility	How easy is it to view and find elements of the music during editing?
Juxtaposabillity	How easy is it to compare elements within the music?
Hidden Dependencies	How explicit are the relationships between related elements in the notation?
Hard Mental Operations	How difficult is the task to work out in your head?
Conciseness	How concise is the notation?
Provisionality	How easy is it to experiment with ideas?
Consistency	Where aspects of the notation mean similar things, is the similarity clear in the way they appear?
Viscosity	Is it easy to go back and make changes?
Role Expressiveness	Is it easy to see what each part of the notation means?
Error Proneness	How easy is it to make annoying mistakes?
Closeness of mapping	Does the notation match how you describe the music yourself?

Table 2. Terms of the Cognitive Dimensions of Music Notations framework used in this paper [14].

example of the terms used in the table, the visibility of the figure is good, showing a clear overview of the entire piece, likewise the juaxtaposability scores highly as the patterns can be compared much more easily than sequences in the score. There are high hidden dependences as the original information has been transformed, with each square representing a smaller amount of information. The simplistic nature of the visualiser scores high on provisionality, consistency and conciseness. The visualisation does not have any meaning unless related to the score, with the underlying notated elements looked up, so has a poor closeness of mapping. Comparing sequences using just the notation would require both hard mental operations, and would be prone to error (error proness), whereas the automated analysis used to build the self-similarity visualisation is easily reproducible and more accurate.

A core concept of visualisation for notated music is its ability to reduce the hard-mental operations arising from manual score analysis [14]. Computer aided analysis also reduces the error proneness of operations. Visualisation can remove un-needed details (filtering [1]) from the score, for example performance markings, therefore improving the conciseness of the results.

Temperley [16] [17] uses visualisation to inform, explain, and evaluate formal analysis by computer. Often using these techniques when analysing a large corpus of music, to immediately show data that would otherwise be difficult to extract from looking directly at the score, or in fact thousands of individual pieces. Temperley also uses these as a way of comparing and refining models for music analysis.

3. INTERACTIVE VISUALISATION

This paper discusses visualisers developed for an original software package (Figure 3), the design of which has been influenced by the seven principles of the VISM (listed in Table 1). In general, it allows different pieces of music in MIDI format, to be opened and visualised quickly, in order to support high provisionality and enable rapid experimentation with analysis techniques. The software can analyse and compare entire corpora or individual pieces, as well as sub-sections or voices (tracks). A historic list of analyses is kept so these can be recalled and modified, retaining low levels of viscosity and commitment, therefore further facilitating experimentation and evaluation (provisionality).

Software and automated analysis has the advantage of processing large amounts of data quickly (compared to manual techniques), but takes considerable amounts of time and care to design and implement. Visualisation tools, such as that described, allow a user to speculatively interrogate data, before committing to more detailed and formal music analysis methods, be they traditional (e.g. Schenkerian) or computer-based (e.g. machine learning see Section 4.4).

4. VISUALISING PITCH

The set of visualisers presented in this section focus on elements of pitch, contour, and melodic interval. Some techniques present the material as overviews of the piece as a whole, others present excerpts in time. For the purposes of discussion and comparison, the majority of visualisations present Bachs Two-part Invention No. 1 (BWV 772) [18], but can be applied to many other examples and genres of music, including non-Western.

4.1 Melodic Contours

A contour representation of music can simply be defined as information about the up and down pattern of pitch changes, regardless of their exact size [19]. Melodic contours are also a key psychological part of music, one that aids the recollection of musical themes [20].

Melodic contours themselves can be illustrated using a score, where it is usually clear in which direction the pitch is going (Figure 4, top). However, once accidentals are introduced (Figure 4, bottom), it becomes less visually distinct. A piano roll (Figure 5) provides a clearer representation of melodic contour. This provides improved closeness of mapping [14], and increases the ease with which sequences can be compared (improving juxtaposability).Piano rolls provide ways for shapes, patterns and contours to be identified. Wood [21] presents related research in which the standard note head is visually modified to show the pitch degree in a more role expressive way, and reports improved speed for sight-reading when compared with standard note heads.

This type of visualisation can also be used to reduce a search space, allowing sequences represented as contours to be visually clustered. The items in Figure 6 show a series of monophonic melodies extracted from Bachs BWV 772. Visually, we can see that the first two patterns are

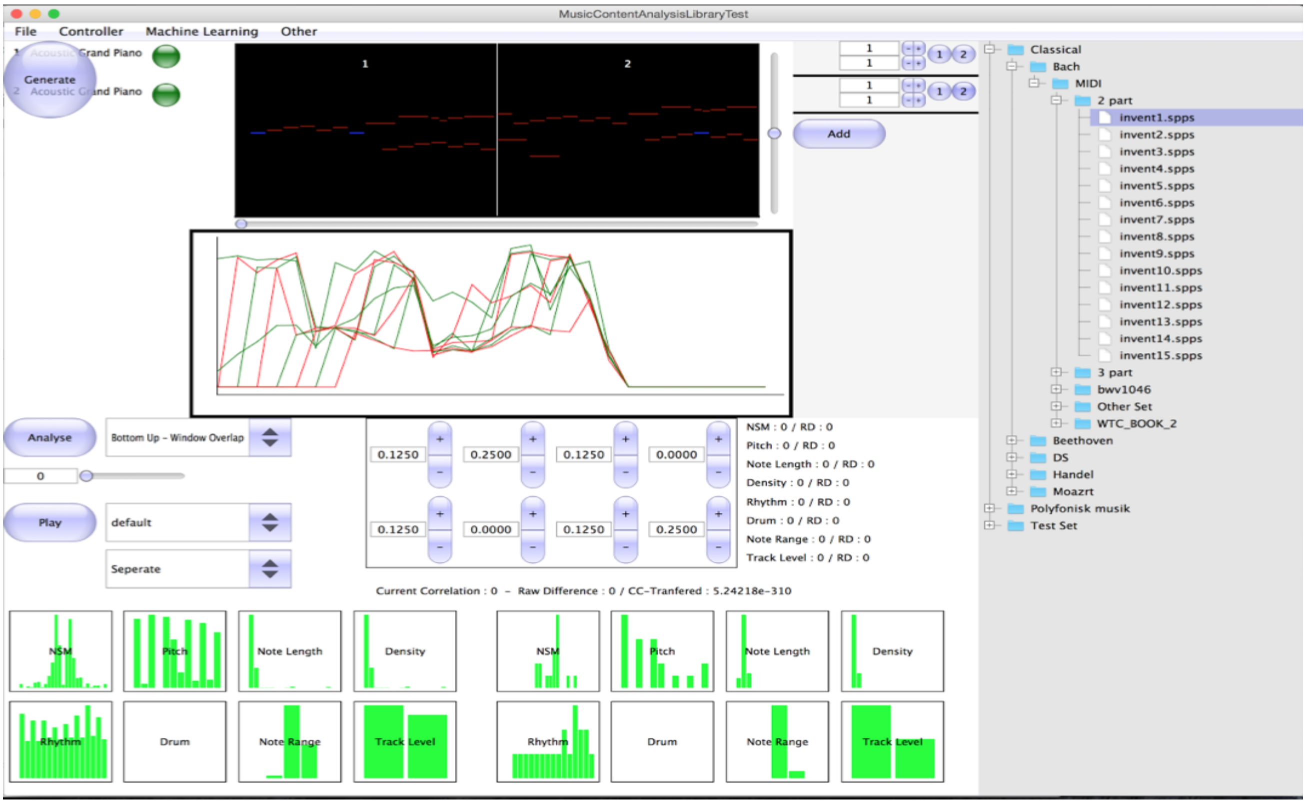


Figure 3. Software created to support visualisation tasks.

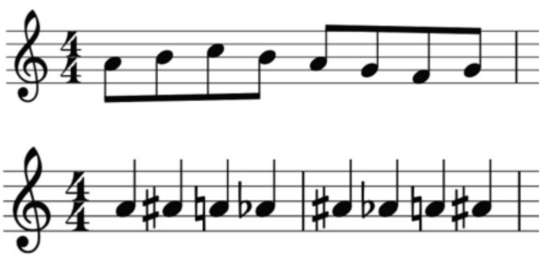



Figure 4. Score with clear melodic contour (top) and obfuscated melodic contour (bottom).

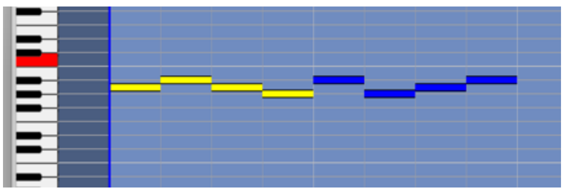


Figure 5. Piano roll representation of Figure 4.

similar, and that pattern 14 is the same pattern inverted. This kind of visualisation allows the viewer to employ the gestalt principles of visual perception, in this case similarity, to group together similar shapes [22]. In this situation the data has filtered out everything but the contour, giving a better overview of the types of contours, which can then be easily related against one another.

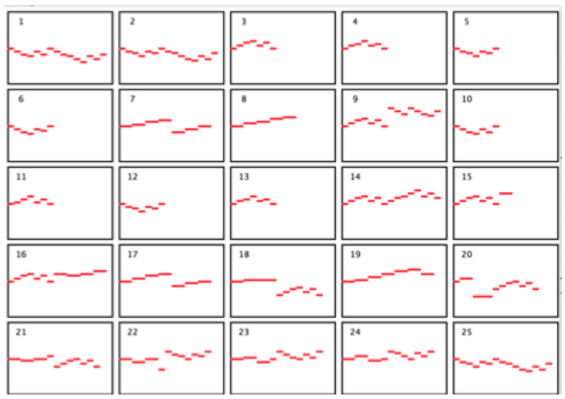


Figure 6. Selection of melodic contours from Bach's BMW 772.

4.2 Intervals

The contour plots provide an overview of the melodic patterns present in the music, but reduce the visibility and role expressivity of the intervals. Temperley [16] uses a histogram of melodic intervals to show the distribution of interval leaps between melodic note sequences within an entire corpus of music material, revealing wider patterns and trends in music. In-so-doing, this hides dependencies in the music, such as the local context and note-to-note relationships (i.e. certain pitches are more unlikely to transition to those depicted in the figure because of their relation to the home key and sensitivity to tonal context). The diagram in Figure 7 shows the interval profile for Bachs BWV 772.

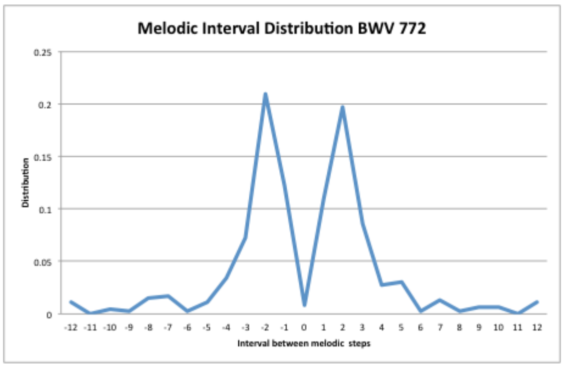


Figure 7. Interval distribution over two octaves in Bach's BWV 772.

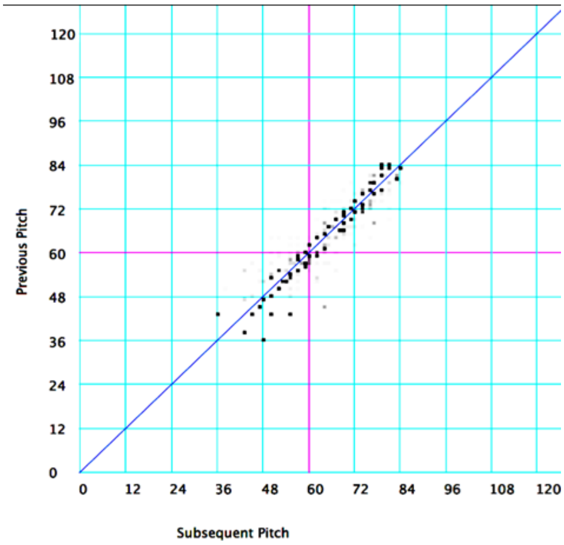


Figure 8. 2D Markov plot of Bach's BWV 772.

A different way to analyse this data, in a way that allows interpretation of pitch, intervals and range, is to use a Markov type model, defining the transition probability between any given notes, in a numeric table format. This, however produces a data table of size 127x127 elements, which is difficult to comprehend in a numeric format, but easily visualised to reveal musical trends and characteristics as illustrated in Figure 10. The design of this once again takes an overview of the data, filtering out the timed elements of the music, to give a detailed overview of the pitch and interval elements. Parts of the plot can be further inspected to reveal exact transition probabilities (details on demand).

From the plot, it can be noted that the intervals in the upper ranges are more likely to jump down in interval, while the opposite effect can be observed in the lower range. Towards the middle the width of the melodic jumps are slightly larger. The blue line along the leading diagonal represents the unison interval (repeated notes), the horizontal deviation from which reflects transitions to subsequent notes. The darker the marker, the more likely the transition. The diagram can also be thought of as a layered series of melodic interval distributions (as in Figure 7), given different starting notes (y-axis).

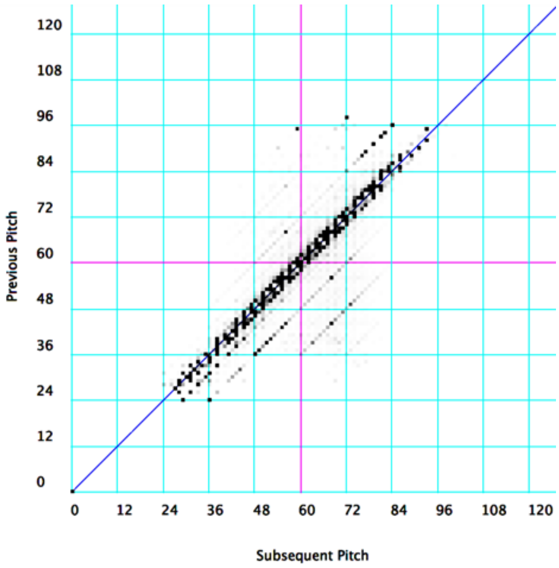


Figure 9. 100 randomly-selected common repertoire Baroque pieces.

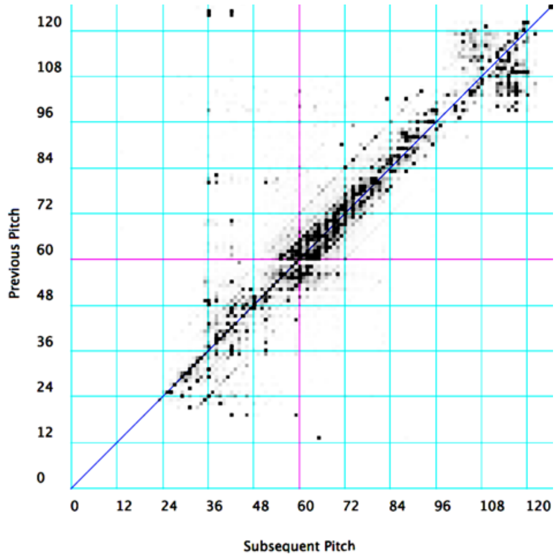


Figure 10. 100 randomly-selected common repertoire Jazz pieces.

Two more plots are shown in this style, but illustrating trends in, and differences between, larger corpora of music: respectively, a collection of 100 pieces of baroque music (Figure 9) and jazz music (Figure 10), selected randomly from a larger corpus. The visualisation process helps to reveal differences between the corpora that would otherwise be harder to discover or articulate. For example, the range of intervals in the jazz corpus is far wider, whereas the baroque is limited to mostly to an octave, and multiples thereof and appears more uniform throughout the range.

4.3 Pitch Distribution

It is instructive to consider pitch usage in general terms. Temperley [16] considers the distribution of pitches within a piece to be an intrinsic element that grounds the overall tonality and key in western music. Key is something that

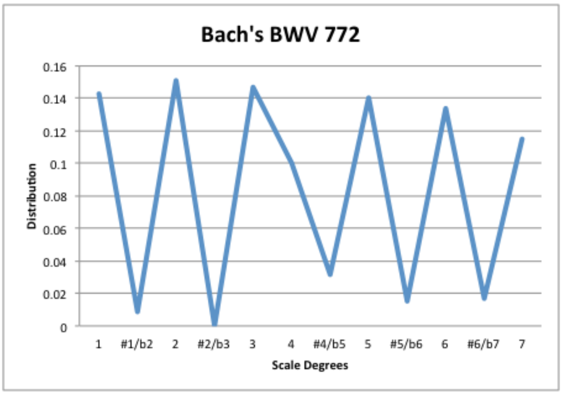


Figure 11. Major Key Profile.

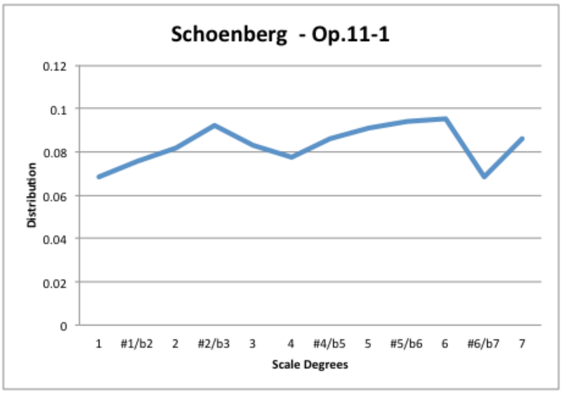


Figure 12. Pitch distribution in Bach's BWV 772.

musicians are trained to detect [16], but for which Temperley has developed automated methods. To illustrate, Figure 11 shows an ideal key profile describing the average distribution of pitches within a piece in C major, which can also be considered a coarse measure of pitch-class appropriateness in relation to key. For comparison Bachs BWV 772 (Figure 12) is also visualised. It is easy to visually infer the similarity of the distribution within the piece (known to be in the key of C) and the generalised representation (Figure 11). Smaller more nuanced details are also visible, such as the fact that the piece, although in C major, has more instances of D than the tonic C. Such details can be enough to fool automated analysis, as detailed in the next section, but things are clearer to the eye.

Other metadata can also loosely be inferred. A less pronounced distribution may indicate a piece that uses several different keys or tonalities beyond the diatonic. Atonal music, such as serialism, may confound such analysis and appear entirely different when visualised, such as Schoenberg Op.11-1 (Figure 13).

4.4 Key

Visualisation can help guide and test formal analysis. For example, a machine learning algorithm was developed that could infer the key based on the pitch profile of a piece. Bachs Well-tempered Clavier (Book 2) [24] was chosen as a test set, as it has two pieces in each of the 24 keys, providing an ordered pattern of tonality.

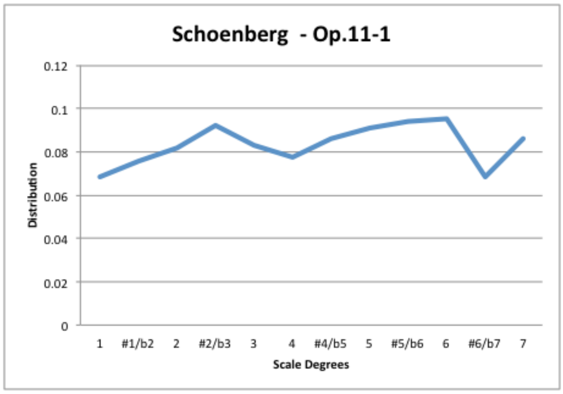


Figure 13. Pitch Distribution in Schoenberg Op.11-1 [23].

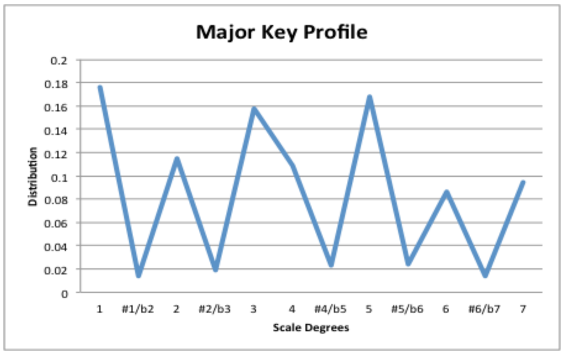


Figure 14. Pitch Distribution for Bach's BWV 870.

Figure 15 presents the detection results of the model, for each piece, ordered by their BWV number. The results of this experiment show that the algorithm is mostly able to predict each of the keys, and the graph can be inspected to find the relative confidence of each prediction as well as identify anomalies and deviations from the expected results. Bachs methodical progression through alternating major and minor keys within the collected work produces a visual pattern in the plot (discernable from the gestalt law of good continuance [22]), the deviations from which identify errors in the key detection model and, in turn, nuances in Bachs approach to key.

The algorithm makes three mistakes, out of a total of 48 predictions, corresponding to the anomalies circled in the figure. In one instance, the algorithm has predicted a key of D minor when the nominal key is C-Major. By visualizing the pitch profile of the piece (Figure 14), using the techniques suggested in Section 4.3 it can be observed that the overall ratio of pitch D, is higher than the tonic and 5th compared with an ideal plot (Figure 11), Indeed, this detection anomaly is attributable to Bachs actual use of D minor (and other keys) in the piece. This indicates a limitation of the analysis technique, in conflating the pitch profile of an entire piece without sensitivity to modulation, but nonetheless raises an interesting musicological question of why this and not other pieces from the set fall foul of this limitation.

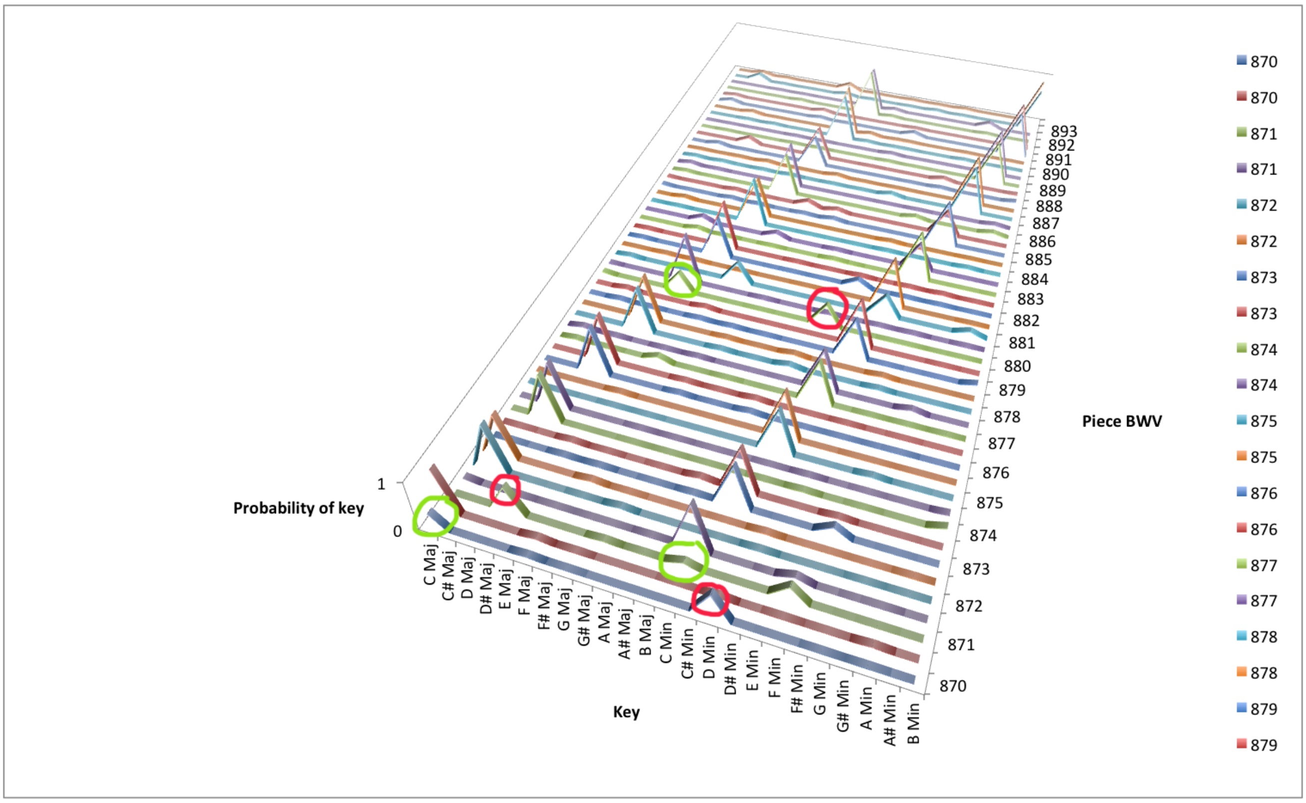


Figure 15. Visualisation of a machine learning algorithms prediction of the 48 pieces of Bachs well-tempered clavier book 2 [24]. The 3 mistakes are BWV numbers 870 part 1, 871 part 1 and 880 part 1. The red highlighting shows the mistakes and the green shows the actual keys.

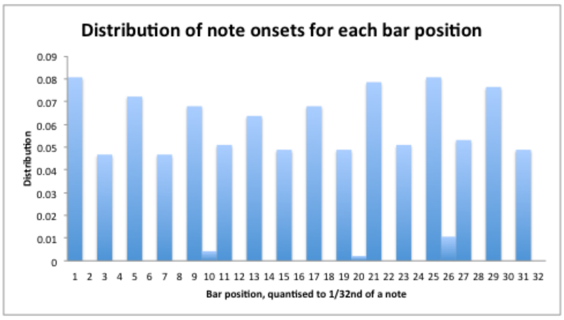


Figure 16. Distribution of rhythm for Bach's BWV 772.

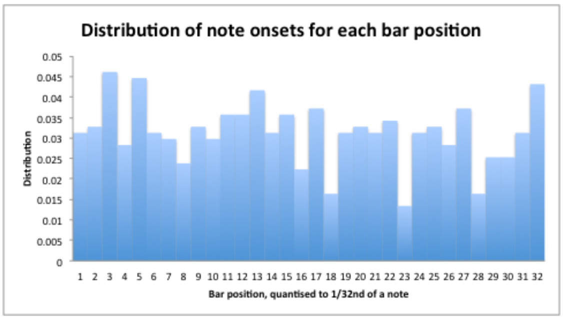


Figure 17. Distribution of rhythm for Beethoven's Op. 53..

5. VISUALISING TIME

Visualisation can also be used to reveal patterns in musical time, as in the case of rhythm, tempo, and density. Time also provides the metrical structure to a sequence of pitches. Taking the Bach piece BWV 772 as before, and visualizing the rhythmic aspects of the piece, several patterns are revealed. The elements under consideration are Note Onset, Note Length, and Density should be merged.

5.1 Note Onset

The basic rhythmic plot, note onset (Figure 16) shows the ratio of note onsets in each position of the bar for the entire piece. The events are first quantised to 1/32nd of a note, to remove noise caused by micro variations in time. The plot shows us, that simpler divisions of the bar are more likely to contain notes than more complex ones, shown by the regular distribution and preponderance of quavers and semi-quavers. The middle of the bar has the least note activity in general, whereas the 1st quaver beat, and 4th quaver beat have the most. Comparing this to Beethovens piano sonata No.21 Op. 53 (Figure 17), a piece from a much later period, shows a complete contrast in the structure, with a much more uniform distribution of note onsets, with the second semi-quaver bar position (3/32) being the most likely place for a note to be played.

5.2 Note Length

Note length visualisation (Figure 18) does not reveal as much information as some other techniques, but confirms

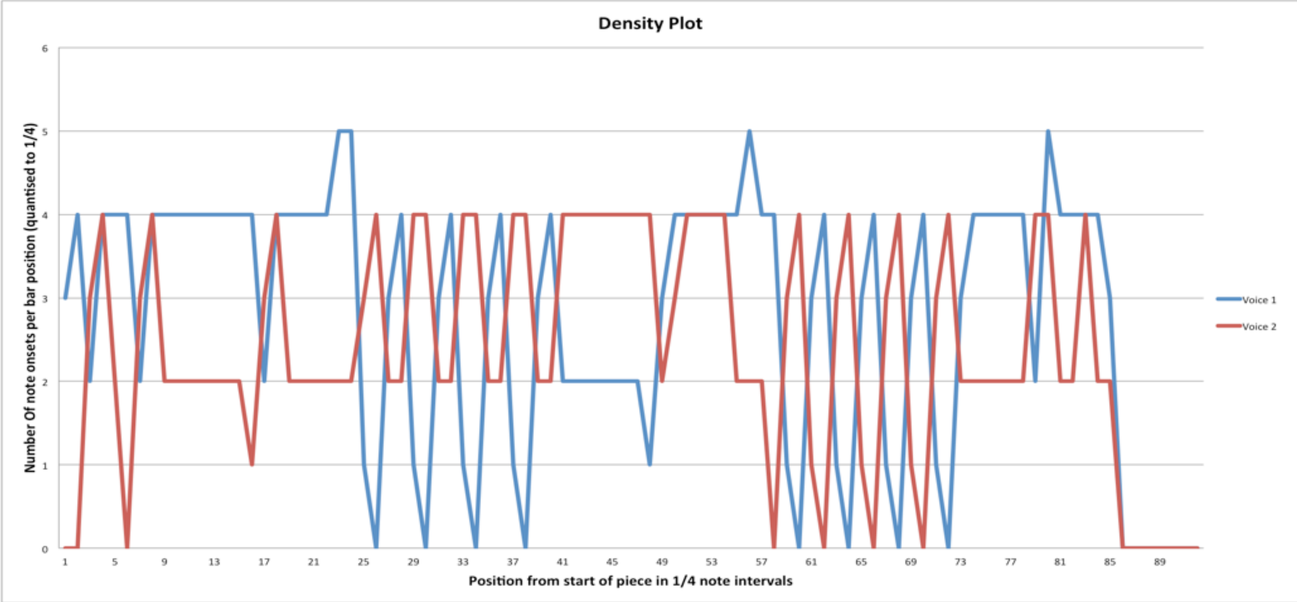


Figure 18. Software created to support visualisation tasks.

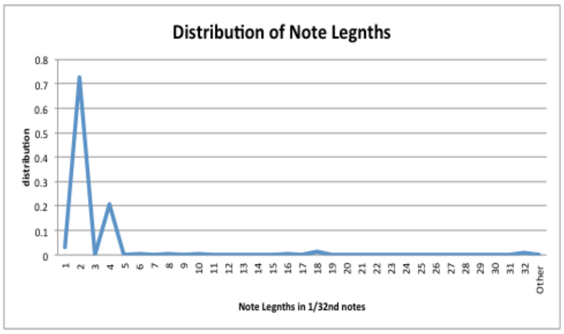


Figure 19. Distribution of note length in Bach's BWV 772.

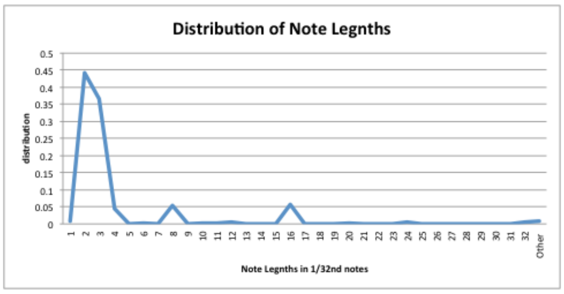


Figure 20. Distribution of note length in Beethoven's Op. 53.

this piece uses mostly note lengths of a semi-quaver in length. Roughly four times as many as using a quaver note. However, comparing this with other examples of music, for example Beethovens piano sonata No.21 Op. 53 (Figure 19), shows for example the use of a dotted semi-quaver (3/32) note length is more common than either a quaver or crotchet, and a value not even used in the Bach piece.

5.3 Rhythmic Density

Rhythmic density can be defined as the number of note onsets that happen during a beat or other window of time. The analysis is computed by calculating the number of onsets in each density window, and plotting the changes over time for each voice (note that only the first 12 measures are shown in Figure 20). Using Bachs BWV 772 again, several repeating patterns are visually observable between the two voices.

Figure 20 shows that only three of 48 windows have both voices indicating a density reading of 4 simultaneously. The sharp peak in Voice 1 at 23-24, is indicated as the most intense, a result of the piece using demi-semi-quavers (see figure 21). From windows 25 to 41, the voices are alternating in a strict pattern. This representation provides a concise overview, but does not differentiate between chords and rapid melodic phrases, reducing the visibility and juxtaposability of data. However, while a finer resolution could reveal more detail, it would also reduce conciseness, with four times as many data points. This represents a common trade-off between the dimensions, as observed in other notations [14].

In general, the techniques discussed in this section show that one method will reveal certain information at the sake of obscuring others, and that sometimes multiple perspectives are needed to fully understand the data.

6. INTERGRATED VISUALISATIONS

Previous sections considered elements of music in isolation, but visualisations can also reveal relationships between different dimensions of music. The ability to integrate musical characteristics and model the complex interwoven principles between them is a prime objective of music analysis and visualisation. The diversity and variety of such interconnections makes this difficult, but it is possible to combine multiple dimensions of characteristics to reveal more complex and interesting patterns.

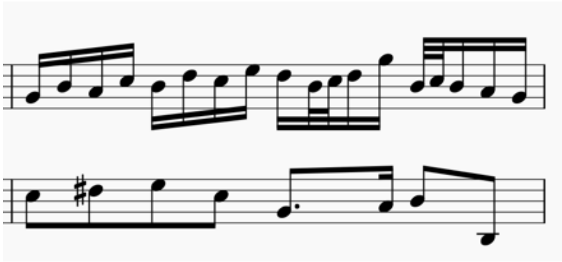



Figure 21. Demi-semi-quavers in bars 6. Relative to points 23-24 on figure 20.

Two related elements of music that can be integrated for visualisation and analysis are rhythm and pitch. A sequence of notes can be considered a pitch change after a given length of time, and it is possible to build up the frequency of these different event combinations and display the result. Given a standard composition the number of options is vast, and represents a complex problem. However, this is relatively easy to visualise (Figure 22) by plotting the change in interval against the difference between note onset, with the colour level (brightness) showing the ratio. In the example (Figure 22), a visualisation of Bachs Brandenburg concerto BWV 1046 [24] is shown, using this method.

Looking at the analysis, it is clear how consistent the timing of the piece is, with most events falling on quaver note divisions. There is some evidence of quaver-triplets as shown between 12 TPQ (Ticks Per Quarter Note or Crotchet) equivalent to a semi-quaver and 24 TPQ (Crochet), with these taking a value of 16 TPQ. Looking at the overall pitch range the widest range of pitch intervals is a note following on a quavers length after the previous note, with events ranging from +24 semitones, to -17 semitones. This is also where the most events are likely to be played, shown by the density of red dots. At the 1 and 2 semi-quaver duration (12 and 24 TPQ) the pitch is more likely to increase, on any value greater than this, the pitch is likely to decrease. At the semiquaver difference, almost all intervals are present, but compare this to longer duration differences, and intervals start to disappear. An interval change of +4 semitones (major 3rd) does not happen following a previous note whose duration was a quaver. This is quite possibly linked to the rules of strict counterpoint, a technique regularly employed by the composer, but further investigation is subsequently required before drawing specific conclusions. Finally, at the 3-semi quaver duration (32 TPQ) interval, a pitch increase is more likely, but at the crotchet level (48 TPQ) a pitch decrease is more likely.

7. CONCLUSIONS

This paper has reviewed a variety of basic music visualisations to demonstrate their utility to reveal implicit details, patterns, and structures in musical phrases, pieces and broader corpora. Although the visualisations have been informally evaluated with reference to the CDMN framework, another way to evaluate the use of visualisation is to establish whether or not it revealed something that was either not known before or complicated to reveal using other methods. As several of these techniques have made such novel observations about musical structure, they can therefore be considered successful.

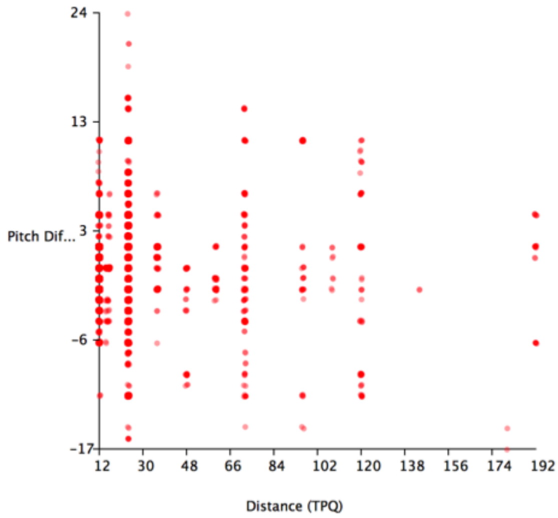


Figure 22. Visualisation of change in interval vs time between note onsets for Bachs BWV 1046. TPQ is defined as the number of ticks per quarter (crotchet) note.

Other further types of studies are also planned in this area, including embedding these visualisation techniques inside music composition software. Such investigations will explore the pedagogical benefits of alternative visual representations of music, looking at how visualisations can inform students understanding of musical process and structure.

Visualisation techniques can also inform the design of generative musical techniques. They allow the identification of characteristics that can become factors of a computer composition models, such as the parameters of a machine learning process. It also allows a degree of quantitative evaluation and comparison between music generated algorithmically and the target musical result. Vickery [25] advocates re-sonifying visualised music representations, formed through analysis of the original music.

While this review of visualisation techniques only scratches the surface of both visual and musical possibilities, it is clear the visual domain can be exploited to provide different perspectives on musical patterns and structures, and make hidden information and insights more accessible to musicians and scholars.

8. REFERENCES

[1] B. Shneiderman, “The eyes have it: A task by data type taxonomy for information visualizations,” in *Visual Languages, 1996. Proceedings., IEEE Symposium on*. IEEE, 1996, pp. 336–343.

[2] B. Craft and P. Cairns, “Beyond guidelines: what can we learn from the visual information seeking mantra?”

in *Ninth International Conference on Information Visualisation (IV'05)*. IEEE, 2005, pp. 110–118.

[3] A. Soriano, F. Paulovich, L. G. Nonato, and M. C. F. Oliveira, "Visualization of music collections based on structural content similarity," in *2014 27th SIBGRAPI Conference on Graphics, Patterns and Images*. IEEE, 2014, pp. 25–32.

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

[5] J. Wolkowicz, S. Brooks, and V. Kešelj, "Midivis: Visualizing music pieces structure via similarity matrices," in *Proceedings of the 2009 International Computer Music Conference, ICMC'09*, 2009, pp. 53–6.

[6] T. Bergstrom, K. Karahalios, and J. C. Hart, "Isochords: visualizing structure in music," in *Proceedings of Graphics Interface 2007*. ACM, 2007, pp. 297–304.

[7] S. Holland, K. Wilkie, A. Bouwer, M. Dalgleish, and P. Mulholland, "Whole body interaction in abstract domains," in *Whole body interaction*. Springer, 2011, pp. 19–34.

[8] D. Jeong and J. Nam, "Visualizing music in its entirety using acoustic features: Music flowgram," in *in Proceedings of the International Conference on Technologies for Music Notation and Representation - TENOR2016*, Anglia Ruskin University. Anglia Ruskin University, 2016, pp. 25–32.

[9] D. Herremans, E. Chew *et al.*, "Tension ribbons: Quantifying and visualising tonal tension," in *in Proceedings of the International Conference on Technologies for Music Notation and Representation - TENOR2016*, vol. 501, Anglia Ruskin University. Anglia Ruskin University, 2016, pp. 8–18.

[10] R. Jackendoff, *A generative theory of tonal music*. MIT press, 1985.

[11] A. Forte and S. E. Gilbert, *Introduction to Schenkerian analysis*. Norton, 1982.

[12] M. Hamanaka, K. Hirata, and S. Tojo, "Implementing "a generative theory of tonal music"," *Journal of New Music Research*, vol. 35, no. 4, pp. 249–277, 2006.

[13] A. Marsden, "Schenkerian analysis by computer: A proof of concept," *Journal of New Music Research*, vol. 39, no. 3, pp. 269–289, 2010.

[14] C. Nash, "The cognitive dimensions of music notations," in *in Proceedings of the International Conference on Technologies for Music Notation and Representation - TENOR2015*, Institut de Recherche en Musicologie. Institut de Recherche en Musicologie, 2015, pp. 191–203.

[15] T. R. G. Green and M. Petre, "Usability analysis of visual programming environments: a 'cognitive dimensions' framework," *Journal of Visual Languages & Computing*, vol. 7, no. 2, pp. 131–174, 1996.

[16] D. Temperley, *Music and probability*. The MIT Press, 2007.

[17] ——, *The cognition of basic musical structures*. MIT press, 2004.

[18] Y. Tomita. (2016) The inventions and sinfonias. [Online]. Available: http://www.music.qub.ac.uk/tomita/essay/inventions.htm

[19] T. Fujioka, L. J. Trainor, B. Ross, R. Kakigi, and C. Pantev, "Musical training enhances automatic encoding of melodic contour and interval structure," *Journal of cognitive neuroscience*, vol. 16, no. 6, pp. 1010–1021, 2004.

[20] R. Aiello and J. A. Sloboda, *Musical perceptions*. Oxford University Press Oxford, 1994.

[21] M. Wood, "Visual confusion in piano notation," in *in Proceedings of the International Conference on Technologies for Music Notation and Representation - TENOR2016*, Anglia Ruskin University. Anglia Ruskin University, 2016, pp. 230–309.

[22] K. Koffka, *Principles of Gestalt psychology*. Routledge, 2013, vol. 44.

[23] imslp. (2016) Schoenberg op.11. [Online]. Available: https://imslp.org/wiki/3_Pieces,_Op.11_(Schoenberg,_Arnold)

[24] D. J. B. Page. (2016) Dave's j.s. bach page. [Online]. Available: http://www.jsbach.net/midi/

[25] L. R. Vickery, "Hybrid real/mimetic sound works," in *in Proceedings of the International Conference on Technologies for Music Notation and Representation - TENOR2016*, Anglia Ruskin University. Anglia Ruskin University, 2016, pp. 19–24.

MELODY RETRIEVAL AND COMPOSER ATTRIBUTION USING SEQUENCE ALIGNMENT ON RISM INCIPITS

Jelmer van Nuss
Utrecht University
j.l.vannuss@students.uu.nl

Geert-Jan Giezeman
Utrecht University
G.J.Giezeman@uu.nl

Frans Wiering
Utrecht University
F.Wiering@uu.nl

ABSTRACT

The RISM A/II database contains metadata and incipits of more than a million compositions. The Monochord search engine can retrieve incipits that are similar to a query using several alignment methods based on pitch raters, weight-based raters and duration-based raters. The performance of all 27 search methods is evaluated using Mean Average Precision metrics and the TREC framework for retrieval performance analysis. The difference in exact pitch between melodies turns out to be the best factor to search with for musical similarity retrieval.

All melodies have metadata such as a composer name, but a portion of the database is labelled as *Anonymus*. A k-Nearest Neighbours algorithm is optimised for the purpose of deanonymisation and used to classify several *Anonymus* songs to test the applicability of this classifier for composer labelling. Using a classifier as a first selection step for deanonymisation purposes turns out to be viable with human correction.

1. INTRODUCTION

The RISM A/II database is a collection of melodies that are stored as incipits, excerpts from the beginnings of notated music in manuscripts collected from libraries, archives, monasteries and schools [1, 2]. This database is not only a useful tool for information look-up on one song, but also to collect similar melodies that give a broader context of the researched melody. As over a million melodies have been stored in this database, having an effective search engine is crucial. While the RISM website [1] has a search function for both metadata and music notation, the power of the search method is limited since it cannot take musical similarity into account. This is why an alternative search engine, Monochord [2], employing more advanced and possibly more accurate search techniques, has been developed.

Monochord is a music retrieval system that is able to find melodies in the RISM A/II database [3]. Monochord compares two melodies by aligning them and calculating a similarity score. The higher this score is, the better the match between the melodies. This search engine has several retrieval methods at its disposal. The performances of these methods have not been researched previously, yet these have to be known before any claims about the performance relative to RISM's search engine can be made.

The goal of this research is twofold. Our first problem is finding an optimal combination of settings that finds similar melodies. Second, after having determined this combination of settings, we use it in an experiment to deanonymise a part of the RISM database.

We begin with an evaluation of the search methods provided by the Monochord search engine. In testing the workings of this engine, we not only gain better understanding of the capabilities and limits of the music similarity retriever, but also gain insights in how to improve the search methods. This part is modelled on previous research conducted by Typke [2], who used similar techniques to create a ground truth set and to evaluate the retrieval results.

Of the 1.148.478 melodies currently stored in the Monochord database, 214.162 have an unknown composer: these are labelled *Anonymus*. Some of these melodies might actually be composed by a composer whose name is impossible for us to retrieve. Others are similar to melodies of which the composer is known. A third type of *Anonymus* songs is that of traditional material that has no single apparent composer. It is desirable to know the true composer of a melody to give credit to the musician, but also to place the works in their context, which may lead to new insights in music history. Using the metadata of similar melodies, we create a classification procedure to determine the composer of the anonymous incipits.

This experiment has a preparation phase and an analytical phase. First off, it is important to understand the mechanisms behind Monochord, how it uses alignment of melodies and several raters to determine melodic similarity, and which aspects differ from the RISM search engine [3]. The retrieval results have to be compared to a ground truth set, which is an expanded version of the one created by Typke [2]. This comparison is made based on precision-recall curves created by means of standard retrieval evaluation tools. The Monochord engine resembles the top k selection used by a k-Nearest Neighbours algorithm. A k-NN model is thus prepared to suggest composer labels for *Anonymus* melodies.

Next, a quantitative comparison of the methods results in the best retrieval method in this experiment. All methods

[1] The RISM database can be queried on: http://www.rism.info/

[2] As an alternative to the RISM search engine, Monochord can be queried on: https://www.projects.science.uu.nl/monochord/risma2/

are used in determining which method is best suited for the purpose of deanonymising melodies using a k-NN. After a quantitative analysis, the best deanonymisation method is qualitatively evaluated by manually checking the plausibility of the composer labels given to *Anonymus* incipits.

2. METHODS

2.1 Pairwise alignment of melodies

Several methods have been researched for modelling melodic similarity. Some examples of these are n-gram methods [4] and geometric methods [5], each with their merits and disadvantages. Alignment of melodies has been implemented before by Kranenburg *et al.*[6]. Their method compares two sequences $\boldsymbol{x}$ and $\boldsymbol{y}$ by taking two symbols from each sequence. These symbols can either be aligned, or there is a gap between the two. Using a substitution score and a gap score, the total alignment score of the two sequences is calculated. This alignment score is to be minimised, as the two most aligned sequences have the least difference in notes and the smallest gaps between two elements of the sequences.

Before alignment, the melodies are transposed using a histogram approach where the pitch shift that maximises overlap in histogram bins is chosen [3]. The Monochord search engine employs these techniques to retrieve similar melodies [3]. All melodies are represented in the `base40` representation.

2.2 Querying with the RISM engine

RISM shows the graphical notation of the incipits to the user, but internally represents these incipits in the Plaine&Easie encoding, with strings such as `'4F8-FA''C/4F8.At3GA'''4C/'F8-F''D/4F` denoting an incipit [7]. A melody is found by first creating a corresponding FAST-index of the Plaine&Easie encoding, where the code is reduced to only pitch values. The engine can then search with or without transposition. In the latter case, only the created FAST-index is used in the search. In the case of transposition, all transpositions of the original search string are added to the FAST-index. This search index is then matched on the existing RISM database [7].

2.3 Querying with the Monochord engine

Potential matches between two melodies are tested for their similarity using a similarity score. This score denotes how well two melodies match, where a higher score is a better match between the melodies. The similarity score of two melodies is calculated during their alignment. The calculation of the scores is where the selected search method plays a role. A search method in the Monochord search engine consists of three types of raters that calculate a subscore by deciding how well the melodies match in the area the rater is specialised in. The sum of these subscores is the overall score that is used for the ranking of the results. The melodies with the highest similarity score will be placed at the top of the list.

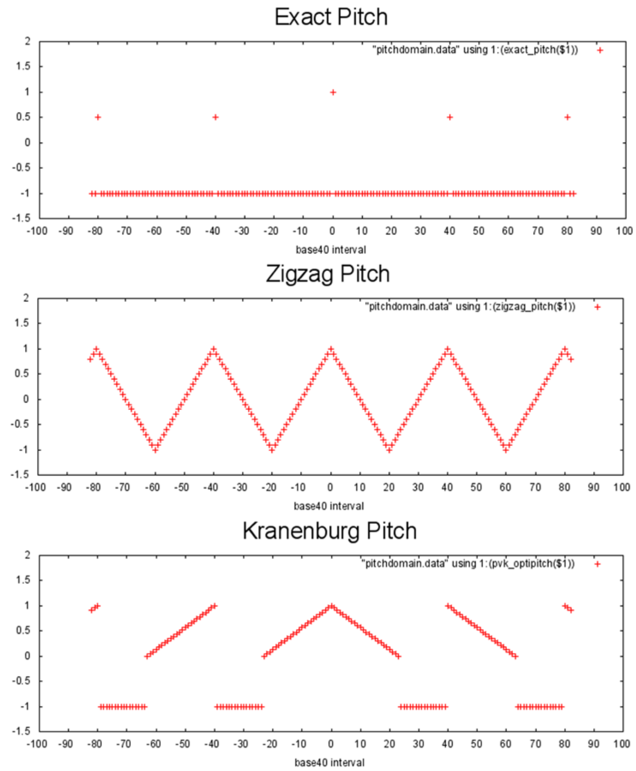


Figure 1. Graphs showing the score assigned to a difference in pitch in base40 representation for the exact pitch (pi2), zigzag pitch (pi3) and Kranenburg pitch (pi1) raters.

Monochord works with search methods that are a combination of three factors with three settings each. First, there is the category of pitch raters that return a value between -1.0 and 1.0. The settings are exact pitch (`pi2`), zigzag pitch (`pi3`) and Kranenburg pitch (`pi1`). The simplest one is exact pitch, which returns a score of 1.0 if aligned notes have an equal pitch, or a score of -1.0 if they differ. A difference of one or more octaves is assigned a score of 0.5. The zigzag pitch rewards notes that are close to each other and punishes notes that differ more. Notes of equal pitch return a score of 1.0. This score decreases linearly to -1.0 when the notes are most different at a distance of 20 in the `base40` representation and then increase to 1.0 where the notes differ by an octave at a distance of 40. The Kranenburg pitch is described by Peter van Kranenburg [6, 8]. For this rater, the score decreases linearly from 1.0 to 0 for intervals up to a fifth; large intervals up to the octaves receive a score of -1.0.

Graphs of the score assignment for each difference in pitch is shown for these raters in Figure 1 [3].

Secondly, there is the category of raters based on metric weight. The settings for this rater is no weight rater (`mw0`), ima weighted (`mw1`) and ima combined (`mw2`). The metric weight for these raters are computed with the inner metric analysis (ima) method by Volk [9]. With the ima weighted method, the influence of a note depends on its metric weight. The weights of the notes of each melody are scaled such that the average weight is 1. The value computed by the pitch rater is multiplied by the average of the weights of the two notes that are compared. The effect is that pitch difference on stressed notes have more influence than differences on less stressed notes. The ima combined method the metric weight has a more independent character than in the previous method. The absolute difference between the two metric weight values is considered and multiplied by the value produced by the pitch rater.

Thirdly, there is the category of duration-based raters. The settings are duration not included (`dur0`), fixed duration (`dur1`) and scaled duration (`dur2`). With no duration, the duration of the notes are not taken into account. With fixed duration, the difference in duration is taken as notated per incipit without any duration scaling. The scaled duration method uses histograms of the duration of notes. We use the duration scaling factor for the query melody that maximises the overlap of the histogram bins [3].

Each search method produces a result file that contains the first 50 ranked search results per query in the ground truth file. A search request consists of a search method and a query ID. Monochord aligns the query melody with all other melodies in the database and calculates the similarity score between the two melodies in a pair based on the search method. A higher score means more similarity to the query and thus a more relevant search result. The resulting melodies are ranked based on their similarity score. All of this is done automatically with a script. A perfect retrieval system will include all the result documents from the ground truth file as the highest ranked retrieved melodies [10]. In practice, we'll encounter melodies that are ranked lower, or melodies that do not appear in the result file at all. Misranking or missing a document affects the performance of a retrieval system.

2.4 Creating a ground truth

The 27 query methods are analysed with a ground truth set [11]. This set contains all relevant result melody RISM signatures per query signature. The retrieval results should show pairs that correspond with those in ground truth: this means that the search method is a good retrieval system. The 2005 MIREX evaluation set of Typke [2] is used for this purpose. Human experts on music were asked to find matches in the 2002 RISM database for a given melody. The participants didn't sift through the whole old database of half a million incipits. Some selective filtering excluded all but 300 incipits per query melody. This filtering was based on for instance large differences in pitch range, duration of the shortest versus the longest note, maximum interval between subsequent notes and editing distance between rhythm strings. This number of incipits was brought down to 50 by manually excluding the remaining incipits that were perceived as too different. Finally, the human experts ranked the 50 incipits based on their similarity to the query melody.

This ground truth data set contains 11 queries with about 10 resulting signatures per query. At the time of construction of Typke's data set, the RISM database contained about half a million melody incipits. The database that powers Monochord has doubled in size since the original ground truth research. It is reasonable to assume that some of the

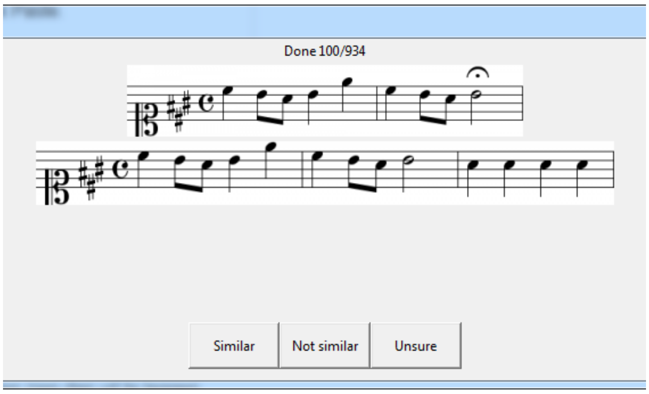


Figure 2. These two melodies are similar and the Similar button should be pressed in this case. Note how the query melody is the beginning of the result melody which has four additional notes.

additions may be truly relevant to one of the ground truth queries. Therefore, this ground truth set needs to be updated before a meaningful analysis of the query methods can be conducted. We update the set by manually checking all query-result pairs that appear in the ranked search results, but not in the original ground truth. These pairs are potential ground truth candidates, because the matching incipit could have been added after Typke's research was conducted. In total 6006 candidate items remain to be cross-checked for similarity by hand.

For this purpose, we create a comparison procedure for the query-result pairs. A computer program splits the candidates in batches of 1001 pairs and shows one pair at a time. The query is shown in musical notation on top, with the result below it. A human evaluator can press one of three buttons to confirm its comparison. The Similar button marks the pair as relevant (a line ending in a 1) and adds it to the ground truth file, then the pair is removed from the program's queue. Figure 2 shows a situation in which the Similar button must be pressed. The Not-similar button stores the pair as not relevant (a line ending in a 0) and removes the pair from the queue. The Unsure button is pressed whenever the evaluator can't make a decision at the moment, for whatever reason, and would like to go on with another pair. The pair is then added to the end of the queue and will return after all other pairs have been checked. The evaluator is shown a new pair after pressing one of these three buttons. Not all images of the musical notation are available on the RISM website, and thus they are unavailable in Monochord. Whenever such a pair comes by, it is handled as Not similar. All pairs without a definitive conclusion are handled as Not similar as well.

Completing one batch takes around 15-30 minutes and is less prone to learning effects that Typke described as possible shortcomings of the experiment [2]. Sequence effects might still occur, as all queries in sorted order. Filtering the pairs as Typke did is not necessary, because checking 6006 pairs manually is feasible. Yet some of the filtering techniques are subconsciously applied, such as rejecting absurdly long incipits or incipits with a greater pitch range instantly.

The ground truth has been expanded by adding 117 new relevant pairs. The new ground truth that is based on the original queries, but with additional results, is published and available for other researchers [3].

2.5 Search method analysis

In evaluating the search methods, we are interested in the Mean Average Precision or area under curve (which are interchangeable terms). The search method with the best Mean Average Precision is designated as the best search method for this incipit database[12, 10, 13]. The methods are not only compared amongst each other, but also relative to an approximation of RISM's innate search engine. A reasonable approximation of RISM's retrieval method is using method `pi2mw0dur0`, as this uses only exact pitch in rating the melodies. This method can be seen as the baseline with which the other methods are compared.

Every search method is tested for precision and recall, which are plotted in the precision-recall curves. An important feature of our TREC files is the ranking of results. This ranking must be used in the evaluation of the search method. Several TREC evaluation tools have been made that utilise ranking (*trec_eval*[14], *trec_eval online*[15, 16], *pytrec_eval*[17]). We use the Python library *pytrec_eval* because the previously written scripts can be transferred to this task.

This evaluation tool requires two types of input: one file containing the expected results (the ground truth) and one file containing the retrieved results. The ground truth file and the result files are stored in the conventional TREC format. The TREC version of the ground truth file is filled with tab-separated lines that contain the RISM signature of the queried document, an iteration number Q_0, its result signature and a relevance rating [18]. Each line has the following format:

$$s_{query} \quad Q_0 \quad s_{result} \quad relevance$$

with $s_{query} \in RISMSignatures$, $Q_0 = 0$, $s_{result} \in RISMSignatures$ and $relevance \in \{0, 1\}$. The result file is similar to the ground truth file, but instead of a relevance rating, it returns a ranking for the document found. Additionally, each line contains a score, representing how well the result matches to the query, and a constant Exp. Both the score and Exp are ignored in this experiment by setting them to zero. A line in the result file has the following format:

$$s_{query} \quad Q_0 \quad s_{result} \quad rank \quad score \quad Exp$$

with $s_{query} \in RISMSignatures$, $Q_0 = 0$, $s_{result} \in RISMSignatures$, $rank \in \mathbb{N}$, $score = 0$, $Exp = 0$.

The evaluation tool uses TREC files, thus we need to convert the information stored in Typke's HTML files to this format. The ranking is not taken into account, all incipits ranked as relevant are used as is. Every melody is referred to with its RISM signature, which is precisely the format needed for our TREC files. For every incipit perceived as relevant with signature s_{result} in a file for a query with signature s_{query}, we create a line

$$s_{query} \quad 0 \quad s_{result} \quad 1$$

where the 1 at the end signifies this pair of query and result is a relevant pair, or a match.

2.6 Deanonymisation of melodies

Once the best-performing search method for retrieval based on melodic similarity has been determined, we can use this method to create data for the deanonymisation classification algorithm. We use a k-Nearest Neighbours algorithm to classify the anonymous melodies. A k-NN retrieves the label for the k elements that are most similar to the element that is to be classified. The most occurring label is said to be the classification of the unknown element. In the case of deanonymisation of melodies, the labels are composer names. As the search results provided by Monochord are ranked from most similar to least similar, we can simply take the top k results as the neighbours and use their metadata to get their composers.

The composer names are available in the RISM database as metadata of the melodies. This forms a mapping between all RISM signatures and their composers or an *Anonymus* label. The correct classifications are thus easily generated: it consists of looking up the melody signature in the mapping and then returning the composer-part in the metadata. If none of the neighbours have a composer label, the classification of the melody will simply be *Anonymus*.

Training and test data for the classification algorithm is widely available. Of the 1.2 million melodies, about a million have a known composer. We randomly sample an amount of incipits with known composer and split the sample 50%-50% in a training and test set.

During the training phase, we use cross-validation to get the best value for k. Here, we use a smaller set of 40 incipits, which is split in a training and test set. The cross-validation consists of testing a k-NN with a certain k on the provided training data. We perform hyperparameter optimisation for k by using a grid search to test all values $k \in \{1, 2, 5, 10, 20, 50, 100\}$ and all $n = 27$ search methods [19, 12]. This takes $\mathcal{O}(k \times n)$ trials and the computation is quite costly, thus we would like to minimise the amount of trials. We first test the k-values only on the best retrieval method and find a good value for k. Then, we use this k to trial all the search methods. Only $\mathcal{O}(k + n)$ trials have to be completed in this manner. The performance of all such k-NNs are compared, after which the combination of k and search method of the best k-NN is selected.

These best k-NN settings are used to initialise the final classifier. A full set of 100 incipits, split in a training and test set, is used for this phase of the experiment. The performance of the classifier is determined using the test data set. It is important to test on a set different than the training set, as overfitting could occur. Overfitting is visible whenever there is great performance on the training set, but poor performance on the new test data. The classifier is stable whenever the performance of the training and test sets is similar.

[3] The revised ground truth is available here: http://www.projects.science.uu.nl/music/resources/

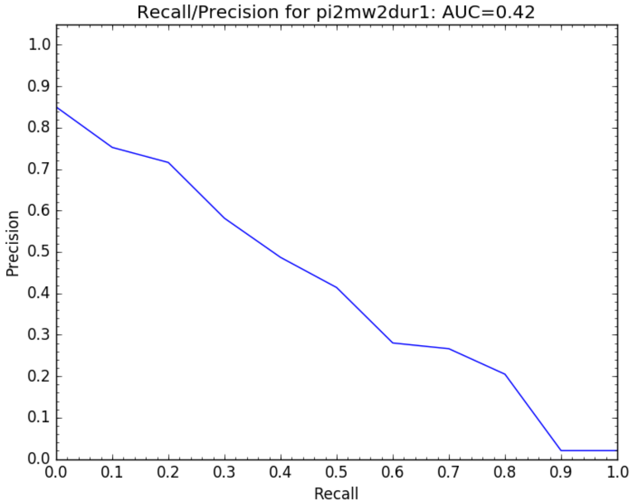


Figure 3. The precision-recall curve for retrieval method pi2mw2dur1. The area under curve, or mean average precision is 0.42, the highest in the series.

This trained classifier can in principle now be used to determine the composer of an anonymous song. This could be done for the 200,000 occurrences, but we randomly sample 100 melodies and evaluate some of the generated labels manually.

3. RESULTS

3.1 Search method analysis

Each of the 27 search methods produces a precision-recall curve from which the mean average precision is calculated. The mean average precision ranges from $0.03 - 0.42$ in the plots. The best method seems to be `pi2mw2dur1` (exact pitch, ima combined, fixed duration) with an area under curve of 0.42. The results are plotted in Figure 3.

Using exact pitch (`pi2`) gives the best results, with an average AUC of 0.38 in a range of $0.31 - 0.42$. The Kranenburg pitch (`pi1`) is the worst performer with an average AUC of 0.21 in a range of $0.03 - 0.32$. The exact pitch curves are plotted in Figure 8(b) in Appendix A, and the Kranenburg pitch as a comparison is shown in Figure 8(a) in Appendix A.

The best duration to use is fixed duration (`dur1`) with an average AUC of 0.33 (see Figure 10(b) in Appendix A). The best use of weight-based raters is by using none (`mw0`) with an average AUC of 0.35 (see Figure 9(a) in Appendix A).

These findings correspond with the best overall method, except for the weight-based rater factor. After a closer look, the method `pi2mw0dur1` seems to be a close runner-up with an AUC of 0.41 (see Figure 4). The overall performance of the ima combined (`mw2`) methods is not that different from `mw0` either, with an average AUC of 0.33.

The baseline approximation of the RISM search engine by using `pi2mw0dur0` results in an AUC of 0.35. Many of the search methods produced worse results than the baseline, but most of the exact pitch family produced equal or

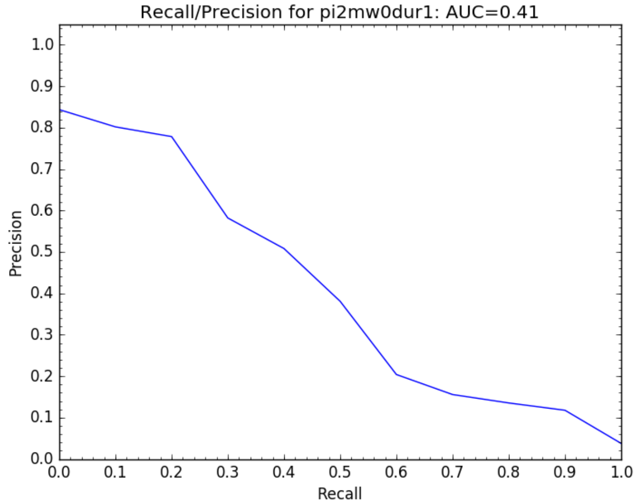


Figure 4. The precision-recall curve for retrieval method pi2mw0dur1. The area under curve, or mean average precision is 0.41, the runner-up in the series.

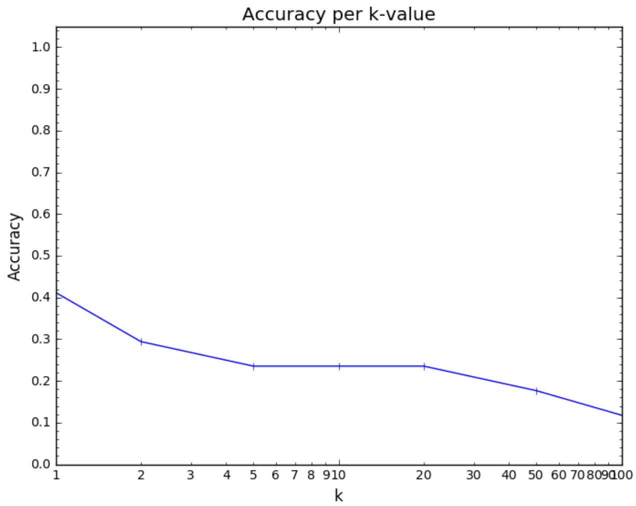


Figure 5. The accuracy of a k-NN for different values of k. The accuracy decreases with an increase in k.

better results.

A full table of AUCs for all search methods is specified in Appendix B.

3.2 Deanonymisation of melodies

Using 40 melodies as training set, we test the seven different values for $k \in \{1, 2, 5, 10, 20, 50, 100\}$ with the best retrieval method `pi2mw2dur1` to gain insight in the effect of the k-value on composer classification accuracy. Removing the *Anonymus* songs based on their ID is fallible process, as the existing set of IDs of known *Anonymus* songs turned out to be incomplete. There are still a few melodies with the *Anonymus* label hidden in the known data set. After filtering these out of the 40 melodies, we are left with 38 incipits.

The accuracy curve in Figure 5 shows that the accuracy decreases as k increases. This seems to have an intuitive reason, as with an increasing k, the share of wrong neighbours also increases. As the most similar songs are placed on the top, a low k will more likely consist of melodies with the wanted composer. The algorithm with a higher k will desperately try to come up with matches at the bottom, even when all the matching pairs have already been found. These bottom suggestions are more likely to be uninteresting, or even counterproductive, for composer classification. And yet the voting power of all these incipits is equal in a k-NN. If some composer turns up in the bottom results often enough, it will overthrow the correct decision made by the top results.

The best retrieval method might not be the best method for composer classification. Therefore another run is performed using $k = 1$ for all 27 search methods.

Method	Accuracy	Method	Accuracy	Method	Accuracy
pi1mw0dur0	0.211	pi2mw0dur0	0.263	pi3mw0dur0	0.263
pi1mw0dur1	0.158	pi2mw0dur1	0.158	pi3mw0dur1	0.158
pi1mw0dur2	0.211	pi2mw0dur2	0.211	pi3mw0dur2	0.158
pi1mw1dur0	0.053	pi2mw1dur0	0.263	pi3mw1dur0	0.158
pi1mw1dur1	0.000	pi2mw1dur1	0.263	pi3mw1dur1	0.053
pi1mw1dur2	0.000	pi2mw1dur2	**0.316**	pi3mw1dur2	0.106
pi1mw2dur0	0.211	pi2mw2dur0	**0.316**	pi3mw2dur0	0.263
pi1mw2dur1	0.158	pi2mw2dur1	0.263	pi3mw2dur1	0.211
pi1mw2dur2	0.158	pi2mw2dur2	**0.316**	pi3mw2dur2	**0.316**

Table 1. Table of accuracies per method, trained on the smaller set of 40 items. Bold numbers signify the highest accuracy.

The methods with the highest accuracy are `pi2mw1dur2`, `pi2mw2dur0`, `pi2mw2dur2` and `pi3mw3dur2` (see Table 1). The best retrieval method `pi2mw2dur1` has the second-highest accuracy, which will therefore also be considered in the possible parameters.

The k-NN is now trained on values for $k \in \{1, 2, 5\}$ and on the methods `pi2mw2dur1`, `pi2mw1dur2`, `pi2mw2dur0`, `pi2mw2dur2` and `pi3mw3dur2`. The data consists of 100 incipits randomly selected from the known melodies. These items are split in a 50% training set and a 50% test set.

The best classifier parameters turned out to be $k = 1$ with the `pi2mw2dur1` method (see Table 2). These settings resulted in a maximum accuracy of 0.375 on the test set.

	k=1	k=2	k=5
pi2mw1dur2	0.354	0.292	0.271
pi2mw2dur0	0.354	0.313	0.271
pi2mw2dur1	**0.375**	0.354	0.313
pi2mw2dur2	0.354	0.316	0.271
pi3mw2dur2	0.354	0.333	0.271

Table 2. Table of accuracies per parameter setting, trained on the full set of 100 items. The bold number signifies the highest accuracy.

Of the 100 melodies, 58 were given a non-*Anonymus* label. The guessed composers of the first eight such entries are given below in Table 3.

Using RISM's search engine [1], we find that incipit 450.202.307-1.1.1 classified as *Sperger, Johannes* indeed contains that name in the list of previous owners of the manuscript. The manuscript was put together by Joseph Michael Zink. This label seems plausible.

Signature	Composer
450.202.307-1.1.1	Sperger, Johannes
851.002.964-1.1.1	Werner, C.
702.020.071-1.1.1	Simonis, Ferdinando
240.006.107-1.1.1	Spohr, Louis
650.007.101-1.1.2	Meyerbeer, Giacomo
500.195.253-1.2.1	Paisiello, Giovanni
150.204.949-1.1.1	Gräfe, Johann Friedrich
454.013.591-1.1.1	Kluger, Johann Florian

Table 3. Table of deanonymised incipits and their composer labels in random order.

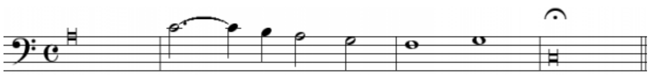



Figure 6. The incipit for 650.007.101-1.1.2, which is the original query (by *Anonymus*).

Incipit 851.002.964-1.1.1 classified as *Werner, C.* contains limited information besides the musical notation, thus a check is impossible.

Incipit 702.020.071-1.1.1 classified as *Simonis, Ferdinando* is called *Les noces?* and is part of a collection of French and Italian songs produced during Simonis' lifespan. The incipit is said to be arranged by the Frenchman André Jean Baptiste Bonaventure Dupont, and its manuscript is stored in Saint Omer's (France) public library. It seems more likely that Dupont is the composer of this song. The label of this incipit is questionable.

Incipit 240.006.107-1.1.1 classified as *Spohr, Louis* is called *Da wir uns niemals wieder finden in B-Dur* and is part of a collection of principally German melodies. The collection originated in 1808, which is during the German Spohr's lifespan. While this contextual evidence seems to make the attribution of this incipit to Spohr plausible, using any of the other search methods shows an abundance of related incipits by Mozart. Indeed, this incipit is a piece by Mozart. This example shows that contextual and music notational inspection are complementary methods of composer attribution analysis.

Incipit 650.007.101-1.1.2 classified as *Meyerbeer, Giacomo* is called *Falsibordoni* in Phrygian mode and is part of a homonymous collection of the same melody in different modes. This collection was put together in 1880, while Meyerbeer died in 1864. Still, it is possible that a piece Meyerbeer wrote was transposed in all modes after his death. Meyerbeer started studying music in Italy in 1816 [20], which explains the Italian name and the collection is stored in the Italian *Archivio diocesano*. The context seems to make the composer classification plausible, but once more music notational analysis shows otherwise. The query for 650.007.101-1.1.2 is shown in Figure 6, while the resulting Meyerbeer composition 452.020.643-1.2.1 is shown in Figure 7 for comparison.

Incipit 500.195.253-1.2.1 classified as *Paisiello, Giovanni* is called *Tenebre e pianto siamo in F-Dur* and is part of a

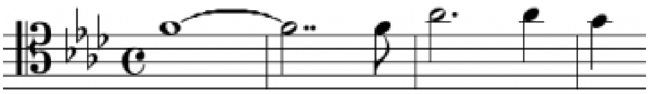



Figure 7. The incipit for 452.020.643-1.2.1, which is the resulting incipit (by Giacomo Meyerbeer). Notice how the incipit differs from the one in Figure 6.

collection of two other Italian melodies, produced in 1770. This is halfway through the Italian Paisiello's life, which makes this attribution plausible.

Incipit 150.204.949-1.1.1 classified as *Gräfe, Johann Friedrich* titled *Ich hab' es oft gesagt in G-Dur* is part of a collection that was produced during Gräfe's life, and his name appears next to one other melody in this collection: *Getrost mein Sinn erheitre dich in F-Dur*. Furthermore, the University of California owns another collection [21] that includes melodies of Gräfe, along with one called *Ich hatt' es oft gesagt in B-Dur*. The RISM ID of this collection is 000.114.155, which indeed gives us the melody with signature 000.114.246-1.1.1 that is an identical, but transposed, copy of our original incipit. This is a confirmed label.

Incipit 454.013.591-1.1.1 classified as *Kluger, Johann Florian* contains limited information besides being a part of a collection with dances exclusively written by Friedrich Joseph Kirmair and Josef Gellert. This collection contains solely German titles and storage locations, while all of the collections in the RISM database including Kluger's works are located in Czech libraries. The only information in favour of this label is the overlap in timespan of the three composer's lives and the 1800-1824 timestamp of the collection. The correctness of this label is questionable.

Three out of eight labels turned out to be quite plausible guesses. This precisely corresponds with the accuracy of the best k-NN classifier found using a test set. This finding makes the classifier results more convincing.

It took a fair amount of time to manually check these labels, but it takes significantly less time than having to come up with an initial guess via human effort. An effective strategy proved to consist of three stages. First, the collection the incipit is from can be scanned for similarities in composers or titles. Next, the timespans of the composer's lives and song publications should correspond. Another strategy is to compare languages, storage locations of the manuscripts, and country of birth or other important locations in the life of a composer. A final (or perhaps first) check is to analyse the matches by musical notation, as this will sometimes conflict with the results found in the contextual analysis. A good amount of music historical knowledge is necessary for this manual effort of label checking.

4. CONCLUSION

The search method `pi2mw2dur1` gives the best melodic similarity retrieval results. The method `pi2mw0dur1` is a good second choice, and might even be preferred when computation cost is factored in, as the ignored factor doesn't need to be calculated. Using exact pitch seems to be much more accurate than any other pitch rater, while the other settings do not matter as much and can be toggled off for an increase in speed.

Whether the best scoring methods are truly nearly equal in results is an interesting topic for further research. We assumed the RISM search engine uses a search method that is equal to `pi2mw0dur0`, it only uses exact pitch, and used that method as a baseline for the other search methods. To provide a true comparison with RISM's search engine, we would have to request the results for our ground truth queries and use this as the baseline. This is an opportunity to make our findings more reliable, yet under our assumption we expect that our claims will remain the same. Another point of improvement would be to look at the complementarity of search methods. Whereas the search methods are analysed in isolation, it might be possible that certain methods are suitable for one type of melody, while another method covers other types. Together, the range of accurate retrievals might be greater than they would be in isolated methods.

The deanonymisation process with a k-NN as described above has an acceptable accuracy (three out of eight plausible labels in the manual check), but the procedure is not accurate enough to become automated. A suggestion for further research would be to check the resulting labels more vigorously, and to do this for more classifications than the eight offered in this paper.

The accuracy of a stand-alone program for deanonymisation of incipits is questionable, but we've shown that using computerised suggestions from classification algorithms can help reduce the manual labour of labelling the songs. The most cost-efficient approach seems to be a combined effort of a computer scientist reducing the search space and offering composer suggestions to a music historian who analyses only a handful of possible composers, instead of the thousands the problem originally started with. For the purpose of giving suggestions, or narrowing the possible composers down to merely a few names, interesting follow-up research would be to test the accuracy of a k-NN that returns multiple labels. Instead of returning the best label, such a k-NN could return the top N composer suggestions. Whenever the true label is in this set of N labels, it is marked as correct. This will result in an equal or higher accuracy as the original k-NN used in this paper (a multiple label k-NN with $N = 1$), as the first result is always the same, with the multiple label k-NN having the benefit of having additional guesses. Such a classifier could conceivably achieve an accuracy that is worth automatising, whose result would be a set of possible composers that the music historian has to inspect for each incipit.

Points for further research include using the manual verifying strategies as features in machine learning applications, such as a k-NN. Perhaps using the collection an incipit is in, the title and composer's language, and the timespans to make a labelling decision can increase the accuracy of deanonymisation classifiers.

This research suggests that improving RISM's innate search engine is worthwhile, as the performance of alternative search techniques was found to be better than the baseline.

Computerised suggestions for composer labels are found to be a promising topic with room for improvement.

5. REFERENCES

[1] RISM, [accessed 2016-07-01]. [Online]. Available: http://www.rism.info/

[2] R. Typke, M. den Hoed, J. de Nooijer, F. Wiering, and R. C. Veltkamp, "A ground truth for half a million musical incipits," *Journal of Digital Information Management*, vol. 3, pp. 34–38, 2005.

[3] Monochord, [accessed 2016-07-01]. [Online]. Available: http://www.projects.science.uu.nl/monochord/

[4] J. Wolkowiczand and V. Keselj, "Analysis of important factors for measuring similarity of symbolic music using n-gram-based, bag-of-words approach," *Advances in Artificial Intelligence Lecture Notes in Computer Science*, pp. 230–241, 2012.

[5] G. Toussaint, "The geometry of musical rhythm," *Discrete and Computational Geometry Lecture Notes in Computer Science*, pp. 198–212, 2005.

[6] P. van Kranenburg, A. Volk, F. Wiering, and R. C. Veltkamp, "Musical models for folk-song melody alignment," *Proceedings of the International Conference on Music Information Retrieval*, pp. 507–512, 2009.

[7] J. Diet and M. Gerritsen, "Encoding, searching, and displaying of music incipits in the rism-opac," *Music Encoding Conference Proceedings, 2013 and 2014*, pp. 11–14, 2015.

[8] P. van Kranenburg, "A computational approach to content-based retrieval of folk song melodies," Ph.D. dissertation, Utrecht University, 2010.

[9] A. Volk, "The study of syncopation using inner metric analysis: Linking theoretical and experimental analysis of metre in music," *Journal of New Music Research*, vol. 37, no. 4, pp. 259–273, 2008.

[10] C. D. Manning, P. Raghavan, and H. Schtze, *Introduction to Information Retrieval*. NY, USA: Cambridge University Press New York, 2008.

[11] J. Urbano and M. Schedl, "Minimal test collections for low-cost evaluation of audio music similarity and retrieval systems," *International Journal of Multimedia Information Retrieval*, vol. 2, no. 1, pp. 59–70, 2013.

[12] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012.

[13] M. B. Kelly, "Evaluation of melody similarity measures," Master's thesis, Queens University, 2012.

[14] Text Retrieval Conference, "trec_eval," 2009, [accessed 2016-07-01]. [Online]. Available: http://trec.nist.gov/trec_eval/index.html

[15] S. Chatzichristofis, K. Zagoris, and A. Arampatzis, "trec_eval online," 2011, [accessed 2016-07-01]. [Online]. Available: http://thetrecfiles.nonrelevant.net/

[16] ——, "The trec files: the (ground) truth is out there," *Proceedings of the 34rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2011.

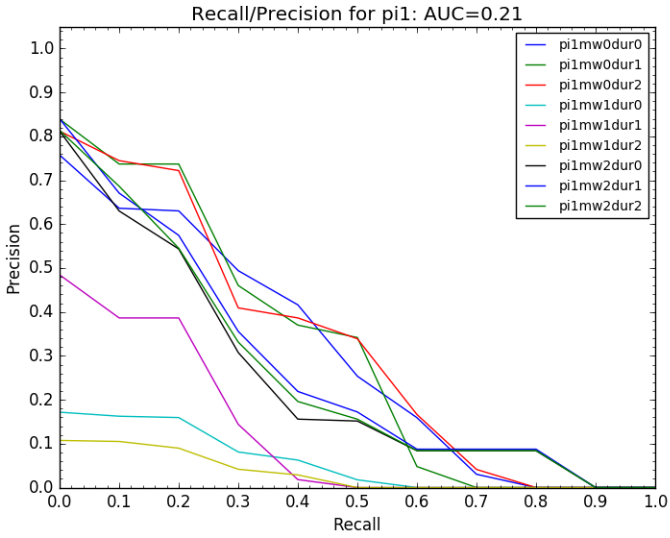
[17] "pytrec_eval," [accessed 2016-07-01]. [Online]. Available: http://www.github.com/XI-lab/pytrec_eval

[18] B. Peddi, H. Xiong, and N. ElSherbiny, "Trec_eval: Ir evaluation," 2010, [accessed 2016-07-01]. [Online]. Available: http://curric.dlib.vt.edu/modDev/package_modules/MidtermModuleTeam5-TRECevalFinal.pdf

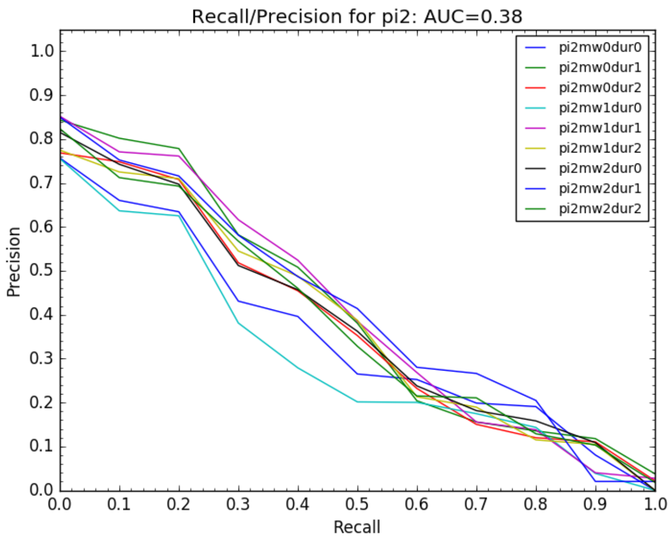
[19] A. B. Hassanat, M. A. Abbadi, G. A. Altarawneh, and A. A. Alhasanat, "Solving the problem of the k parameter in the knn classifier using an ensemble learning approach," *International Journal of Computer Science and Information Security*, vol. 12, no. 8, pp. 33–39, 2014.

[20] H. Becker, *Giacomo Meyerbeer in Selbstzeugnissen und Bilddokumenten*. Reinbek: Rowohlt Verlag, 1980.

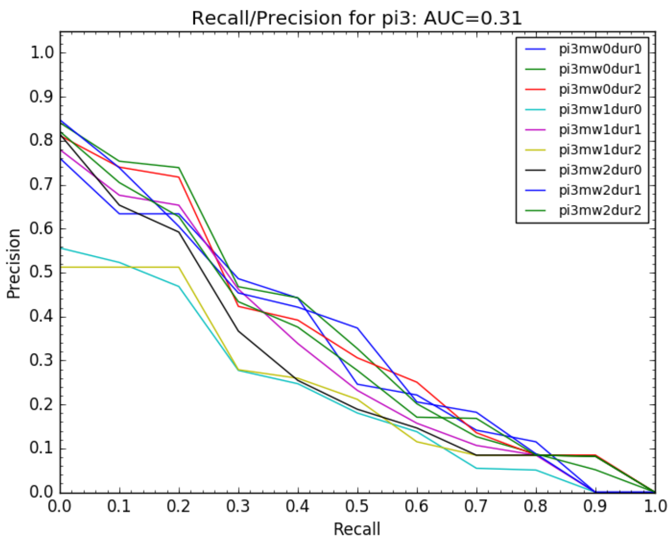
[21] "University of California collection," [accessed 2016-06-25]. [Online]. Available: http://beta.worldcat.org/archivegrid/collection/data/830325643



(a) All retrieval methods using Kranenburg pitch. The average AUC of the methods is 0.21.



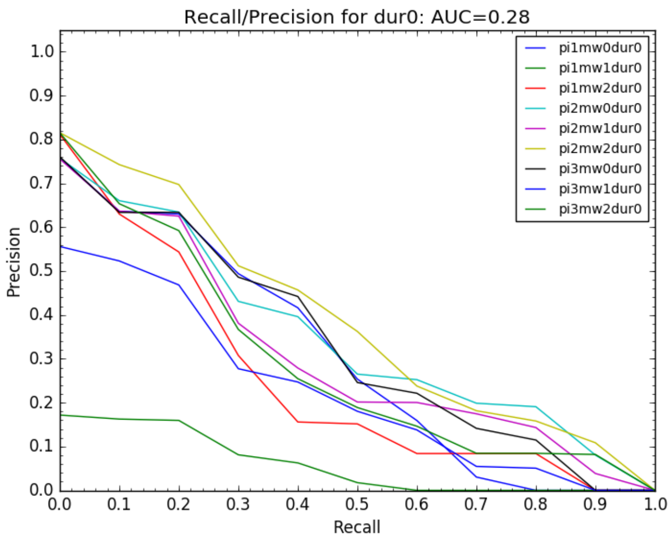
(b) All retrieval methods using exact pitch. The average AUC of the methods is 0.38.



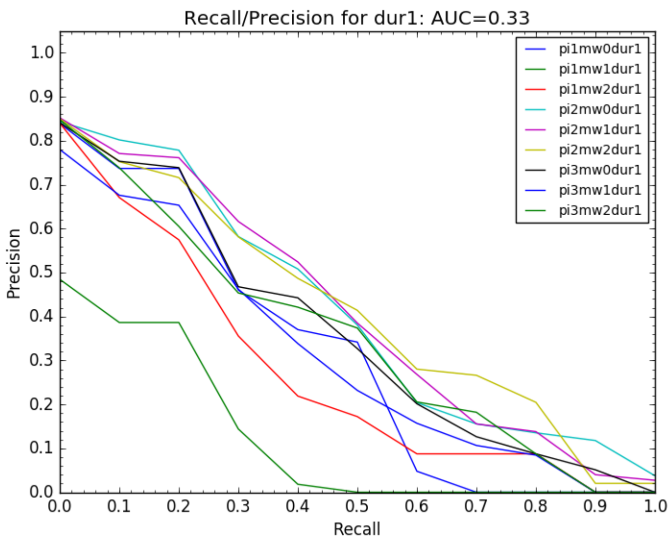
(c) All retrieval methods using zigzag pitch. The average AUC of the methods is 0.31.

Figure 8. The precision-recall curve for all methods from the pitch rater family. The average AUC of the methods is shown in the titles.

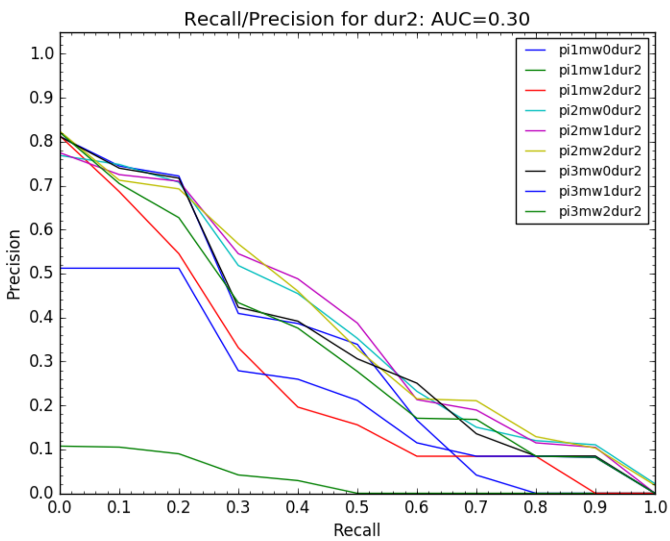
A. PRECISION-RECALL CURVES PER SETTING



(a) All retrieval methods using no duration rater. The average AUC of the methods is 0.28.

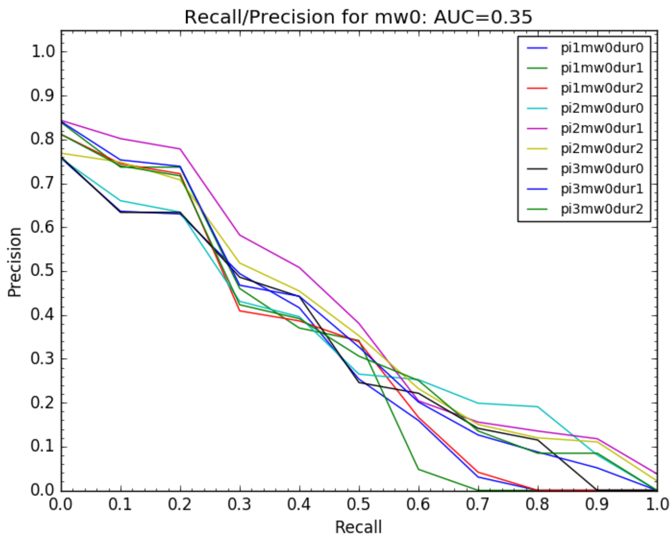


(b) All retrieval methods using the fixed duration rater. The average AUC of the methods is 0.33.

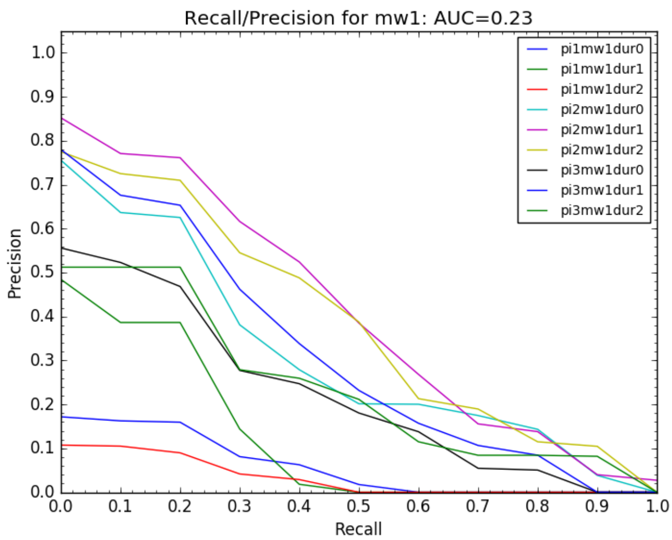


(c) All retrieval methods using the scaled duration rater. The average AUC of the methods is 0.30.

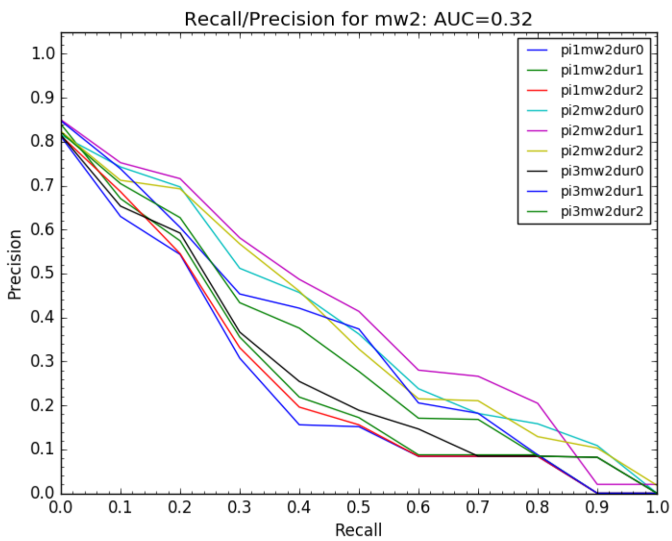
Figure 10. The precision-recall curve for all methods from the duration rater family. The average AUC of the methods is shown in the titles.



(a) All retrieval methods using no weight-based rater. The average AUC of the methods is 0.35.



(b) All retrieval methods using the ima weighted rater. The average AUC of the methods is 0.23.



(c) All retrieval methods using the ima combined rater. The average AUC of the methods is 0.32.

Figure 9. The precision-recall curve for all methods from the weight-based rater family. The average AUC of the methods is shown in the titles.

B. AUC TABLE FOR THE SEARCH METHODS

Method	AUC	Method	AUC	Method	AUC
pi1mw0dur0	0.30	pi2mw0dur0	0.35	pi3mw0dur0	0.33
pi1mw0dur1	0.31	pi2mw0dur1	0.41	pi3mw0dur1	0.36
pi1mw0dur2	0.32	pi2mw0dur2	0.38	pi3mw0dur2	0.35
pi1mw1dur0	0.06	pi2mw1dur0	0.31	pi3mw1dur0	0.22
pi1mw1dur1	0.12	pi2mw1dur1	0.41	pi3mw1dur1	0.31
pi1mw1dur2	0.03	pi2mw1dur2	0.39	pi3mw1dur2	0.24
pi1mw2dur0	0.24	pi2mw2dur0	0.39	pi3mw2dur0	0.29
pi1mw2dur1	0.27	pi2mw2dur1	0.42	pi3mw2dur1	0.35
pi1mw2dur2	0.26	pi2mw2dur2	0.38	pi3mw2dur2	0.33

Table 4. Table of the AUC for each of the search methods.

FORMALIZING QUALITY RULES ON MUSIC NOTATION – AN ONTOLOGY-BASED APPROACH

Samira Cherfi, Fayçal Hamdi, Philippe Rigaux
CNAM
firstname.lastname@cnam.fr

Virginie Thion
IRISA
virginie.thion@irisa.fr

Nicolas Travers
CNAM
nicolas.travers@cnam.fr

ABSTRACT

We address the issue of expressing and evaluating quality rules on music notation. Since music engraving is a highly flexible process that can hardly be constrained by universal principles and rules, score production still heavily relies on the user expertise in order to make context-dependent decisions. We therefore propose a quality management approach based on a formal modeling of this expertise. We show how to use such a model to express context-aware rules that can be evaluated either *a priori* to prevent the production of faulty notations, or *a posteriori* to assess quality indicators regarding a score or a corpus of scores. The paper proposes a simple ontology for musical notation, shows how quality rules can be formally stated and evaluated, and illustrates the approach with examples drawn from a large digital library of scores.

1. INTRODUCTION

Music production is now strongly assisted by sophisticated and powerful computer softwares. They allow to combine all the elements of the notation language, and can in most cases make appropriate decisions regarding quality-sensitive aspects such as, e.g., layout or spacing. We could therefore expect that computer technology would guarantee the production of high-quality scores, validated with respect to a set of well-accepted engraving principles that constrain the notational language [1].

1.1 Quality issues

However, it is well-known that this language is highly flexible, due to many cultural and historical contexts where each one presents their own idiosyncrasies. There is an inherent, context-dependent freedom in the adjustment of the common graphic and symbolic elements that constitute a specific score, and this prevents the enforcement of even the most widely accepted principles which can turn out to be inappropriate in some specific cases. To take one example, transcribing a manuscript, in particular for ancient music, raises trade-off issues between the necessity to preserve the original intent of the author, and the adaptation of handwritten notation to the custom knowledge of today's performers.

The authors of the present paper have been confronted with the need to address issues related to the consistent production of high-level quality corpora encoded in XML-based formats (i.e., MusicXML [2] or MEI [3, 4]), and had to deal with the poor support offered by existing tools. The current, ad-hoc solution adopted so far is to publish a booklet of *editorial rules* prior to the production of the corpus scores. They often take the form of a textual, informal document that enumerates guidelines regarding the encoding of music and helps the editor(s) to find a consistent approach balancing the need to both preserve the sources and to deliver a consistent material to users (musicologists, performers, librarians) who access the corpus.

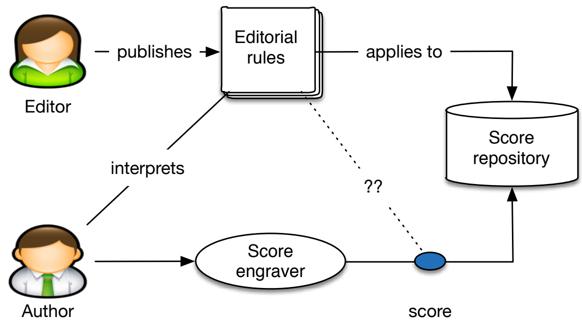


Figure 1. Editorial rules

The approach is not fully satisfying. As shown by Fig. 1, there is no direct nor formal association between the rules and the encoded scores. The link depends on both the interpretation of a user, and the specific features of the score engraver. Even though we assume that the scores are edited by expert authors, keen to comply with the recommendations, nothing guarantees that they are not misinterpreted, or that the guidelines indeed result in a satisfying encoding. Moreover, rules that are not backed up by automatic validation safeguards are clearly non applicable in a collaborative context where un-controlled users are invited to contribute to the collections.

1.2 Formalizing editorial rules

Editorial rules are based on two important assumptions. First, they assume that both the editor and the author share a common *expertise* on music notation, and that this expertise supports rules, conveyed by sentences in natural language whose meaning is expected to be unambiguous.

Second, the author is assumed to "process" the rules while creating a new score, and guarantees that the resulting encoding fulfills them.

In the present paper we propose to formalize these assumptions, in such a way that *expertise*, *rules*, and *rules fulfillment* can all be explicitly stated and automatically validated. The main components of the approach are summarized by Fig. 2. Its foundations consists of an ontology of music notation, representing the concepts and domain-specific knowledge. Rules can be expressed (by an editor, possibly helped by experts) as formal sentences built from these concepts, and validation can be carried out by a reasoner that, given an instance of a score (interpreted as an instance of the ontology concepts), checks the rules fulfillment.

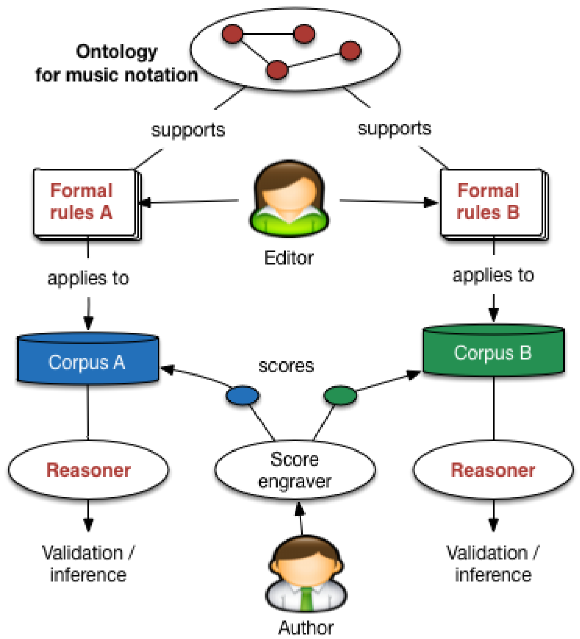


Figure 2. Formalization of rules

The rules might differ from one corpus to another, e.g., there is no reason to assume that the same set of constraints hold for a corpus of Renaissance music and for the Complete Works of Anton Webern. This approach therefore allows to specialize the definition of what a correct engraving is in a specific context, and can be seen as a complement of Finale, Sibelius or MuseScore that deliberatley aim at proposing full-featured, non specialized engraving options.

The rest of the paper intends to demonstrate the promising perspective brought by associating a sophisticated encoding of music notation (say, using the MEI format) with knowledge-based management tools. We use as a driving motivation the expression and control of editorial rules on music scores corpora. Section 2 gives an overview of the approach, along with background notions. Section 3 proposes a simple ontology, and Section 4 examples of editorial rules. We discuss how the methodology can be used in a broader perspective in the concluding section.

2. BACKGROUND AND MOTIVATION

Let us first develop why, in our opinion, current technology falls short to support quality assessments on score encoding. We then provide some background on the field of formal ontologies and reasoning, and explain how this field can be used in the context of music notation.

2.1 Dealing with quality issues

The flexibility of music notation is such that it is difficult to express and check quality constraints on the representation that would universally hold. For instance, many formats we are aware of do not impose that the sequence of notes/rests in a measure exactly covers the measure duration defined by the time signature. As another example, in polyphonic music, nothing guarantees that the parts share common signatures and durations. So, even with the most sophisticated encoding, we may obtain a score presentation that does not correspond to a meaningful content (the definition of which is context-dependent), and will lead to an incorrect layout (if not a crash) with one of the possible renderers.

Besides, scores are being produced by individuals and institutions with highly variables motivations and skills. By "motivation", we denote here the purpose of creating and editing a score in a digital format. A first one is obviously the production of material for performers, with various levels of demands. Some users may content themselves with schematic notation of simple songs, whereas others will aim at professional editing with high quality standards. The focus here is on rendering, readability and manageability of the score sheets in performance situation. Another category of users (with, probably, some overlap) are scientific editors, whose purpose is rather an accurate and long-term preservation of the source content (including variants and composer's annotations). The focus will be put on completeness: all variants are represented, editor's corrections are fully documented, links are provided to other resources if relevant, and collections are constrained by carefully crafted editorial rules. Overall, the quality of such projects is estimated by the ability of a document to convey as respectfully as possible the composer's intent as it can be perceived through the available sources. Librarians are particularly interested by the searchability of their collections, with rich annotations linked to taxonomies [5]. We finally mention analysts, teachers and musicologists: their focus is put on the core music material, minoring rendering concerns. In such a context, part of the content may be missing without harm; accuracy, accessibility and clarity of the features investigated by the analytic process are the main quality factors.

Finally, even with modern editors, qualified authors, and strong guidelines, mistakes are unavoidable. Editing music is a creative process, sometimes akin to a free drawing of some graphic features whose interpretation is beyond the software constraint checking capacities. A same result may also be achieved with different options (e.g., the layer feature of Finale), sometimes yielding a weird and convoluted encoding, with unpredictable rendering when submitted to another renderer.

2.2 Knowledge formalization with OWL ontologies

One of the major achievement of the Semantic Web initiative [6] is the development of OWL, a language to represent *ontologies*.[1] An ontology is a set of axioms and rules that provide formal statements about the concepts (or "classes") and concept occurrences (or "individuals") of some knowledge domain. For instance, `Note` is a basic concept, which can be represented by a class in an OWL ontology, and some A4 in a score is an occurrence of the concept which can as well be represented in the ontology as an individual.

OWL supports inference mechanisms that derive new facts from those explicitly present in the ontology. As a trivial example, since A4 is a `Note`, which itself is a sub-class of `Sound`, a reasoner can infer that A4 is a `Sound`.

Ontologies have been recognized as an essential component for representing knowledge. An ontology commonly agreed to in a given domain constitute an essential basis to express formal statement that represent some domain knowledge, and to build sound reasoning and inference mechanisms related to this knowledge. The formalization of ontologies and reasoning also allows to automatically and safely validate facts, rules and constraints. As such, it constitutes an invaluable support to make sense to massive amounts of semi-structured data that would otherwise be hardly interpretable. While the initial purpose of semantic web technology is the mastering of Web data, its use has now spread to highly specialized knowledge domains. We make the case here for applying this approach to music notation.

The ontology proposed here is formalized using the fragments of OWL 2 [7] corresponding to the description logic $\mathcal{SROIQ}(\mathcal{D})$ [8]. The use of OWL 2 is privileged because it provides a high expressiveness allowing semantic reasoners to verify the consistency of data, to derive new knowledge or to extract information already present. In addition, rules can be added to the ontology to express complex knowledge and provide more inference possibilities. A language of choice is SWRL, the Semantic Web Rule Language [9], which is brefly introduced in Section 4 along with rules examples.

2.3 Ontology-based quality assessment

Axioms and rules that compose an ontology can be used to assess the quality of music notation, assuming the latter is represented in some structured format (e.g., Kern, MusicXML, MEI, etc.). We can then interpret the content of a score in terms of the ontology concepts (see Fig. 3 for an illustration). This helps to reduce the conversion of notation elements from a score into *facts* representing concepts occurrences such as, e.g., "in this voice, in this measure, and for this duration, we find *this* chord." The set of facts that we obtain together represent the notational knowledge encoded in the score, and we can then confront this knowledge to rules that state what are the fair facts.

In formal terms, new facts are produced and a reasoner can check if the ontology, augmented with facts and rules,

[1] https://www.w3.org/TR/owl-features/

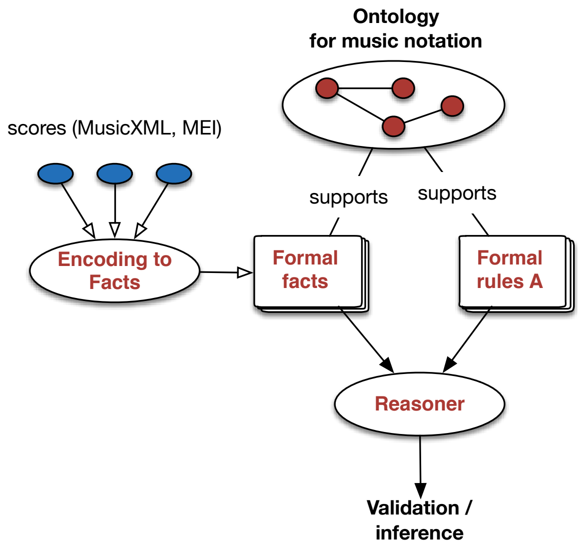


Figure 3. Representing a score as ontology facts

is still consistent and hence provide an information about the notation quality (e.g. accuracy, correctness, etc.). For instance, a fact which states that an event is at the same time a lyric and a rest, introduces an inconsistency when the ontology contains an axiom that says that an event is a disjoint union of these two classes. As another example, a rule stating that time intervals of two different events can't overlap, helps to detect imprecisions in the expression of intervals related to voice events.

As illustrated by Fig. 3, the only non-standard component in this validation process (assuming a well-accepted ontology of music notation) is the converter that takes some score encoding as input and produces facts (usually encoded in RDF) as output. Implementating such conversion is definitely easy, and this makes the approach a quite attractive one, given its potential benefits.

3. THE ONTOLOGY

We now present a concrete application of the above principles, based on an ontology of music notation specifically designed as a support for expressing quality constraints. Some preliminary words of caution are here in order.

3.1 Goals and Restrictions

The popularity and advantages of ontologies led to their usage in managing musical information. We can find hight level or meta data oriented ontologies to manage metadata about musical works [10]. For more content oriented usage, Raimond and al. proposed the Music Ontology [11] to manage basic information about musical works and artists. The objective is to integrate musical works in the Semantic Web and the ontology is consequently used as a base for many music-oriented web services. A similar work is the Kanzaki Ontology [12], a music vocabulary which describes classical music and performances. These ontologies describes music at a work level and are suitable to describe general informations such as music categories

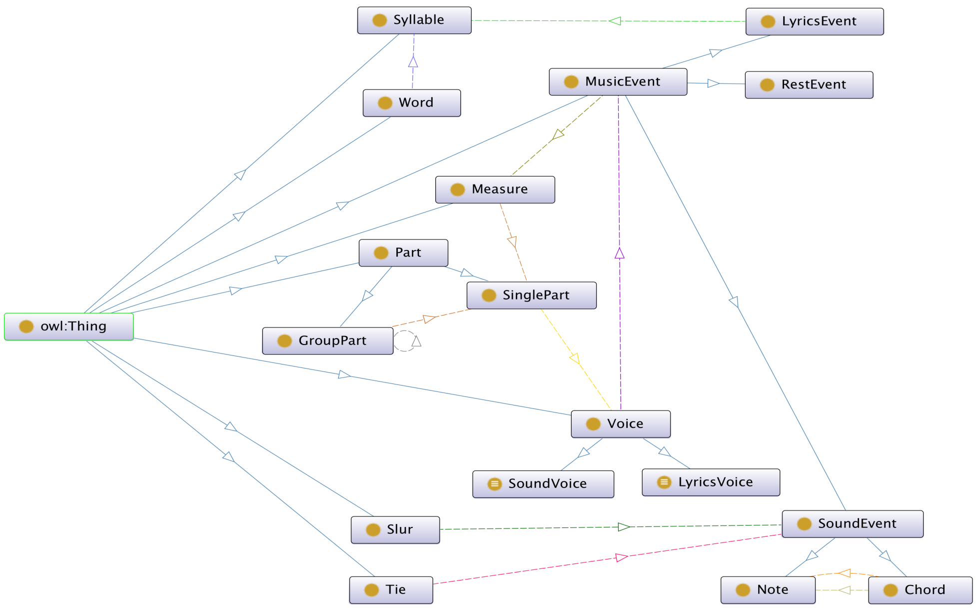


Figure 4. The music notation ontology (OWL modeling)

(Chamber music, Choral music etc.) and performances (musicians, instruments etc.). To handle musical content descriptions, to enhance and facilitate their sharing among communities of both novices and experts, there is a need for more content oriented ontologies. We do believe that such a model would be of invaluable help to let the community formalize discussions and proposals and address issues related to the topic. We hope that the present work, although quite limited in its scope, can serve as an encouragement to initiate such an endeavour.

The proposed ontology is by no means intended to cover the whole knowledge of music notation throughout ages; this would be an extremely ambitious task (at least for the paper's authors) which, at the very least, would require a long, collaborative process. The part of music notation that we aim at modeling here deliberately ignores issues related to the graphical layout of score. This aspect is major in estimating the quality of a score, as witnessed by the countless recommendations that can be found in reference such as [1]. However, it also constitutes a part of quality assessment which can hardly be evaluated from the encoding found in MusicXML or MEI formats. For the sake of simplicity and validation of our approach, we therefore chose to focus on the part of the notation that relates to "music content" in the following. Separating content from layout is not trivial, and to the best of our knowledge there does not exist a common agreement on this issue. We do not pretend to solve it here, but used the intuitive distinction between layout and content as a guideline to support the following decisions:

- All pure graphic instructions: paper size, margin, fonts, glyphs and positioning coordinates, are not considered.
- Directions regarding the assignment of voices and parts on staff are also ignored; this include the clef and textual annotations associated with staves.

This essentially lefts elements that organize the music content are parts, parts in voices and voices as sequences of events. This is elaborated next.

3.2 The MusicNote Ontology

Fig. 4 shows the main concepts of our ontology [2]. The figure is produced by the *Protégé* editor [3]. The explanations that follow should make the major features clear event to non-experts. Essentially, a score is modeled as a *hierarchical structure*, where leaves consist of *voices*, and inner nodes of *parts*. A voice is a sequence of *events*, occurrences of the abstract class `Event` which is refined in several sub-classes. We detail first the structural aspect, then the voice representation.

3.2.1 *Structural aspects*

Let us explain the structural aspect first by taking as an illustration the sketch of a piano concerto score (Fig. 5).

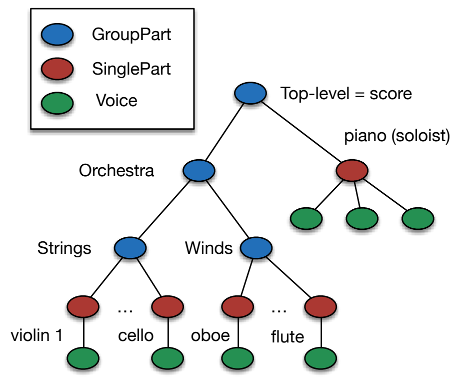


Figure 5. Structure of a score (`GroupPart` and `SinglePart` concepts)

[2] http://cedric.cnam.fr/isid/ontologies/files/MusicNote.html
[3] http://protege.stanford.edu/

A score is made of *parts*. Class `Part` represents an abstract concept which is refined in two sub-concepts:

1. `GroupPart`. A *group* (of parts) consists of a set of subparts, and mostly serves the organisational aspect of the score. For instance (Fig. 5), the orchestral material of a concerto score typically defines a group for wind instruments, another one for string instruments, etc.

2. `SinglePart`. A *single part* encapsulates the music notation elements assigned to an individual performer (instrument or vocal). Fig. 5 shows for instance a single part for the soloist (piano), another one for the violins, cellos, etc.

The content of group part may actually consist of an heterogenous association of single and group parts, as illustrated by the top-level node of Fig. 5 that associates a group (the orchestra) and a single part (the soloist). This is reflected in the modeling of the ontology.

Note that we do not explicitly introduce a *score* concept. In our model, a score is simply the root of the tree of parts, and everything it contains. Such a concept would however be useful to introduce score-level metadata (composer, title, etc.) that would come as siblings of the parts hierarchy. Since we focus on the music content representation, we safely keep the model simple.

3.2.2 *Core concepts: events and voices*

A single part is a container for the core elements of music notation: events and voices. An *event* denotes the production of a noise artifact during a specific time period, called the *duration* of the event. Note that duration in this context has an absolute meaning, and corresponds to an open time interval fully contained in the temporal coverage of the score. The event concept can be refined based on the nature of the produced noise: it can be a sound (`SoundEvent`), text or syllables to be sung (`SyllEvent`) or a silence (). The `SoundEvent` concept itself is decomposed as follows:

- `Note` denotes a simple, non-decomposable sound that can be represented by the well-known attributes `pitch`, `octave` and `accidental`. A more radical choice would be to simply represent a note by its frequency.

- `Chord` is an event composed of at least two notes that all share the same duration.

The status of `RestEvent` is debatable. A rest can be interpreted as an absence of event for a certain duration, and, in a radical perspective that would try to forget the idiosyncratic aspects of music notation, there is *a priori* no need to supply such a concept. One could also argue that rests are first-class notational objects that deserve to be explicitly represented. We can probably find contexts where a half rest is more appropriately represented as two quarter rests. A true, complete modeling of music ontology would have to carefully examine such cases in order to reach a large agreement.

A *voice* is a sequence of events whose durations do not overlap. A voice extends over a time range that can (optionally) be decomposed as a *sequence* of *measures*. A property of a measure is the *time signature*, the value of which can (extreme case) vary from one measure to another.

In summary, a score can essentially be seen, in our model, as a synchronization of an unbounded number of parts, each defining an internal organization of a finite time range split in measures.

3.2.3 *A full example*

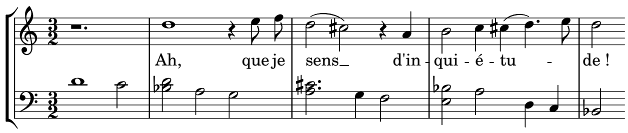


Figure 6. A full example

Let us consider as a full example the score shown in Fig. 6, and its modeling. It consists of two parts, lets' call them "vocal" and "accompaniment". The vocal part consists (in our modeling) of two voices, the first one (called "sopr") composed of sounds, and the second one ("lyrics") of syllables (note that there is no one-to-one rythmic correspondence between syllables and notes, as some syllables cover several notes). The second part consists of a single voice, "bass". The structure is summarized by Fig 7.

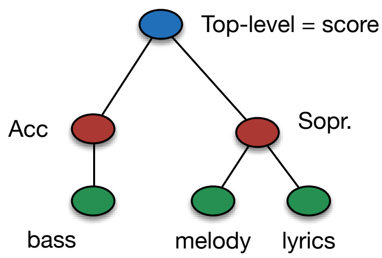


Figure 7. Structure of the example score

Consider now the details of each voice (summarized by Fig. 8). Voice "sopr" is a monophonic voice, instance of `SoundVoice`, each event being either a single note or a rest. Voice "lyrics", instance of `LyricsVoice`, consists of syllables. Finally, voice "bass", instance of `SoundVoice`, contains a few complex events, instance of `Chords`.

This example shows the main feature of how we can interpret a score notation as fact stated with respect to the ontology context. Those facts can automatically be extracted from the MusicXML or MEI encoding, represented in a convenient form (typically as RDF triples) and sent, along with ontology and rules, to a reasoner that will determine the consistency of the whole. Among other motivations, this can serve as a setting to validate quality rules, as discussed in the next section.

$$v_{sopr}(t) = \begin{cases} \perp, & t \in [0,12[\\ D5, & t \in [12,20[\\ \perp, & t \in [20,22[\\ E5, & t \in [22,23[\\ F5, & t \in [23,24[\\ D5, & t \in [24,28[\\ C\#5, & t \in [28,32[\\ \perp, & t \in [32,34[\\ A4, & t \in [34,36[\end{cases} \quad v_{lyrics}(t) = \begin{cases} \perp & t \in [0,12[\\ \text{Ah}, & t \in [12,20[\\ \perp, & t \in [20,22[\\ \text{que}, & t \in [22,23[\\ \text{je}, & t \in [23,24[\\ \text{sens}, & t \in [24,32[\\ \perp, & t \in [32,34[\\ \text{d'in}, & t \in [34,36[\end{cases} \quad v_{bass}(t) = \begin{cases} D4, & t \in [0,8[\\ C4, & t \in [8,12[\\ < B3es, D4 >, & t \in [12,16[\\ A3, & t \in [16,20[\\ G3, & t \in [20,24[\\ < A3, C4is >, & t \in [24,30[\\ G3, & t \in [24,32[\\ F3, & t \in [32,36[\end{cases}$$

Figure 8. Voices as sequences of events (measures 1 to 3)

4. QUALITY RULES

Rules express constraints that music scores should respect. A language of choice to express rules is SWRL [9] which is briefly introduced first. We then enumerate some of the quality rules that can be expressed in this OWL+SWRL framework and conclude the section with few examples.

4.1 SWRL

SWRL is a language that allows to express rules that take the form of an implication: a *body* is a list of statements which are interpreted as true or false depending on the context, and *head* is a statement whose truth value is inferred from the evaluation of the body. It is basically similar to rules in Datalog [13]. Let's take a simple example: the following rules define a `MajorChord` as a `Chord` with three notes `a`, `b`, `c` such that the interval between `a` and `b` is a major third (4 semi-tones) and the interval between `b` and `c` is a minor third (3 semi-tones) [4] .

```
Chord(?x), hasNote(?x, ?a), hasNote(?x, ?b),
           hasNote(?x, ?c),
           gap(?a, ?b, 4), gap(?b, ?c, 3)
    -> MajorChord(?x)
```

Symbols of the form `?x` denote variables. The interpretation of this rule is essentially: if we can find an instantiation of the variables `x`, `a`, `b` and `c` such that the body of the rule is evaluated as true, then we can infer that the head is true as well, i.e., `x` is an occurrence of the new, intentional concept `MajorChord`.

This simple example shows the kind of reasoning that allows to produce new knowledge about a set of facts (taken from a score encoding), and given a modeling of the domain supplied by a generic ontology. The `MajorChord` can now be reused just as any other concept, and we can thus build sophisticated reasoning chains that can be evaluated by a reasoner on a score or a corpus of scores. Let us examine the application of this idea for quality assessment.

4.2 Quality dimensions

Quality measures are commonly organized according to the following quality dimensions [14]: *accuracy*, *completeness*, *trust* and *consistency*. We give below, for each dimension, some possible examples of quality rules for music notation.

[4] For the sake of illustration, the example rely on obvious simplifying assumptions.

Accuracy measures in what extend data values correspond to their considered correct representation. Classically, two kinds of accuracy are considered: the syntactic accuracy and the semantic one. The syntactic accuracy measures the adequacy of data to its expected format. A typical syntactic accuracy rule could check that (AccR1) each note is an existing one (roughly speaking in the domain {C, D, E, F, G, A, B, C}), or that (AccR2) a voice nomenclature is respected, for instance with voices in the domain {Superius; Cantus; Altus; Contratenor}, or that (AccR3) at most one syllable is associated with a note.

The semantic accuracy measures the closeness of a value to a considered true real-world value. Its measurement supposes that there is somewhere a reference for the content to be checked, namely a business expert knowledge or another source to compare to. A syntactic accuracy rule could check that (AccR4) the birthdate associated with each compositor corresponds to the birthdate of a trusted other internal or external given source (e.g. Wikipedia if considered as trustable enough).

The *Completeness* measures in what extent the score contains all the required information, concerning data and metadata. A syntactic completeness rule could check that (CompR1) a figured bass is present, or that (CompR2) at least one syllable is associated with each note, or that (CompR3) each measure is complete according to the figured bass, or (CompR4) the presence of some meta-data.

The *Trust* dimension concerns the trust-worthiness of each dataset, for instance by (TrustR1) checking the provenance information and the confidence in the provider.

The *Consistency* measures the adequacy of data to semantic rules. Such semantic rules may concern any element of the music score. A consistency rule could check that (ConstR1) each note can be played by the instrument (or voice) it is associated with, or that (ConstR2) the musical instruments were created before the compositor date of death.

4.3 Rules expression

Rules such as those above can be expressed with SWRL according to the ontology defined in Section 3. For instance, the rule (AccR1) may be expressed by the following formula.

$$Note \equiv \{A\} \sqcup \{B\} \sqcup \{C\} \sqcup \{D\} \sqcup \{E\} \sqcup \{F\} \sqcup \{G\} \quad (1)$$

As another example, the following rule specializes (ConstR1) by stating that if a measure ($?m$) includes several notes ($?e1$) and ($?e2$) played at the same time ($?i1 \; =?i2$) then it belongs to a part ($?pt$) to which is associated a polyphonic instrument ($?inst$).

$$\begin{aligned} Part(?pt) \wedge hasInstrument(?pt, ?inst) \\ \wedge Measure(?m) \wedge hasPart(?m, ?pt) \\ \wedge SoundEvent(?e1) \wedge SoundEvent(?e2) \\ \wedge differentFrom(?e1, ?e2) \\ \wedge hasMeasure(?e1, ?m) \wedge hasMeasure(?e2, ?m) \\ \wedge during(?e1, ?i1) \wedge during(?e2, ?i2) \wedge equals(?i1, ?i2) \\ \Longrightarrow Polyphonic(?inst) \end{aligned} \tag{2}$$

As shown by the complexity of this last rule, this requires either a close cooperation between a domain expert (e.g., a musicologist, a librarian) and a OWL/SWRL expert, or advances interfaces that let users build their own rules and control their meaning. This constitutes therefore both an exciting and promising axis for interdisciplinary research.

5. CONCLUSION

We presented in this paper an approach that aims at manipulating the content of music notation at a high level of abstraction, using concepts, knowledge and rules that leverage traditional encoding formats. The proposed methodology relies on OWL / SWRL, and we outlined the main steps: formal domain modeling with an OWL ontology, production of facts from the content of MusicXML or MEI documents, expression of rules, and production of new facts and knowledge thanks to a reasoner.

The work presented here is in progress, and is intended both to demonstrate to the TENOR community what can potentially be achieved with techniques that, as far as we know, have not yet been investigated in the music notation domain, and to encourage feedback or direct participation. Building an ontology requires all kinds of expertise, and aims at reaching the largest possible agreement. The present proposal is a step in this direction.

We are currently implementing a platform that focuses on quality evaluation rules. This is motivated by practical needs (we maintain on-line cooperative corpus for which quality issues are a primary concern). This restriction also makes investigations and experiments easier. We expect to be able to demonstrate the platform features during the conference, and hope that it will encourage discussions with the TENOR participants beyond notation quality issues.

Acknowledgments

This work is partially supported by the GioQoSo project [5], and the ANR MuNIR project [6].

6. REFERENCES

[1] E. Gould, *Behind Bars*. Faber Music, 2011.

[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] J. Riley and C. A. Mayer, “Ask a Librarian: The Role of Librarians in the Music Information Retrieval,” in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2006.

[6] T. Berners-Lee, J. Hendler, and O. Lassila, “The Semantic Web. A New Form of Web Content that is Meaningful to Computers will Unleash a Revolution of New Possibilities,” *Scientific American*, vol. 284, no. 5, pp. 34–43, May 2001.

[7] P. Hitzler, M. Krötzsch, B. Parsia, P. F. Patel-Schneider, and S. Rudolph, “OWL 2 Web Ontology Language Primer (Second Edition),” World Wide Web Consortium, W3C Recommendation, December 2012. [Online]. Available: http://www.w3.org/TR/owl2-primer/

[8] I. Horrocks, O. Kutz, and U. Sattler, “The Even More Irresistible SROIQ,” in *KR*, 2006, pp. 57–67.

[9] I. Horrocks and P. F. Patel-Schneider, “A proposal for an owl rules language,” in *In Proc. of the Thirteenth International World Wide Web Conference (WWW 2004*. ACM, 2004, pp. 723–731.

[10] L. Dodds. (2004) Musicbrainz metadata vocabulary. [Online]. Available: http://www.ldodds.com/projects/musicbrainz/schema/mb.html

[11] Y. Raimond, S. Abdallah, M. Sandler, and F. Giasson, “"the music ontology",” in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2007, pp. 417–422.

[12] M. Kanzaki. (2003) Music vocabulary. [Online]. Available: http://www.kanzaki.com/ns/music

[13] S. Abiteboul, R. Hull, and V. Vianu, *Foundations of Databases*. Addison-Wesley, 1995.

[14] C. Batini and M. Scannapieco, *Data Quality: Concepts, Methodologies and Techniques*, ser. Data-Centric Systems and Applications. Springer, 2016.

[5] http://gioqoso.irisa.fr/
[6] http://cedric.cnam.fr/index.php/labo/projet/view?id=41

SMARTVOX – A WEB-BASED DISTRIBUTED MEDIA PLAYER AS NOTATION TOOL FOR CHORAL PRACTICES

Dr Jonathan Bell
Composer
Paris, France
belljonathan50@gmail.com

Benjamin Matuszewski
CICM/musidanse EA1572, Université Paris 8,
STMS Lab IRCAM-CNRS-UPMC
Paris, France
benjamin.matuszewski@ircam.fr

ABSTRACT

The present paper describes the features and implementation of *SmartVox*, [1] an application designed to help vocal ensembles learn and perform polyphonic music.

Technically, *SmartVox* is a distributed web application that delivers audiovisual scores through the performer's mobile devices. From a singer's point of view, this setup allows for the synergy between visual and acoustic stimuli, which facilitates the interpretive and performative processes, particularly in polyphonic passages. It also enables spatial separation of the performers (*cori spezzati*), and speeds up the learning process of unfamiliar musical materials (e.g. microtonal tuning, texts in a foreign language).

The ubiquity of smartphones makes such a distributed system affordable and allows the use of *SmartVox* in multiple contexts, from professional ensembles to pedagogical and recreational practices.

Introduction

Related Work

Many composers in the twentieth century have designed audio systems in order to conduct performers. The first extensive use of technology-assisted conducting dates back to the 1920s, when click tracks were used in order to sync the orchestra to silent movies. [2] In the domain of experimental music, the first and most explicit research is found in Emmanuel Ghent's *Programmed Signals to Performers, a New Compositional Resource* [1], where the author proposed a system derived from the click track technique. This system, the *Coordinome*, was able to send audio signals pre-recorded on a magnetic tape individually to each performer.

During the second half of the century, many systems were developed to guide performers with multiple click tracks in order to realize complex polytempo pieces (Elliot Carter's *2nd String Quartet*, Brian Ferneyhough's *Mort Subite*, Karlheinz Stockhausen's *Helicopter Quartet*, Iannis Xenakis' *Persephassa* to name just a few).

More recently, Phillipe Kocher developed a technology-assisted conducting system for the realisation of polytempo networks [2]. Similarly, the Latvian composer Rytis Mazulis used in many pieces equivalent systems, sending to performers extremely slow glissandi generated electronically [3].

Figure 1. *Au Commencement* (J. Bell, 2016), performed by the *Mangata* ensemble using *SmartVox*.

Performance-Centrism

In most of the above-mentioned cases, technologies of assisted conducting were created to solve *composition-centered* issues: concerned with the realisation of a specific musical material. This focus on composition, to the detriment of its *performative* counterpart, draws attention to difficulties that can be encountered by the person who plays/sings the music. Indeed, the use of click tracks (audio) or animated notation (visual) changes the traditional relationship established between composer, conductor, and performer. In a musical ensemble, these forms of extended notation often imply the replacement of a human conductor by a technological solution. The improvements in precision that accompany the use of such systems do not come without drawbacks. For example, polytempo and microtonal passages become simple to realize, even with musicians placed around the audience, but the stimulation of the player by a live conductor is not guaranteed any more. The challenge for the composer therefore consists of find-

[1] https://github.com/belljonathan50/SmartVox0.1 – the term *SmartVox* was coined by Laurence Brisset from the *De Caelis* Ensemble.

[2] The invention of the click track is usually credited to Max Steiner, although other sources have attributed it to Scott Bradley or Carl Stalling.

ing ways for these new forms of notation to remain as efficient as more conventional setups. According to this *performance-centric* point of view, the singer's reception, appropriation and feedback play an essential role in the assessment of the application.

1. AUDIO-SCORES AND ANIMATED NOTATION

SmartVox is characterised by the use of audiovisual (i.e. multimedia) notation. This recent development was initiated by preliminary research on *audio-scores* [4] and *animated notation* (screen-scores). [3]

1.1 Audio-Scores

Microtonality and polytempo are two realms of investigation which often lead composers to extend their notation with an acoustic element (such as click track, sine-tones, or any other auditory information) [1][2][3][4]. These signals are generally sent through an earpiece, so that they only concern the performer and not the audience. Such audio cues can be particularly helpful for vocalists. Indeed atonality and microtonality impose even more challenges to singers than to instrumentalists, because singers cannot rely on gestural automatisms to find their pitch and adjust their intonation to others. Writing microtonal music for *a cappella* voices, or placing musicians in the performance space therefore present difficulties that audio-scores attempt to solve, using audio technologies as a means to 'augment' the traditional notation. This compositional research was initially motivated by the intuition that the coupling of auditory and notational media would give the performers access to new and expressive situations, thus investigating the fields of computer-aided composition and computer-aided performance. [4]

In comparison to traditional sheet music, an audio-guide is often initially surprising for the singer, but quickly becomes a very useful cue which one can rely on. The performer is not only asked to reconstruct his part from the symbols he reads, but also to imitate what is heard through the earpiece. The notational input is therefore not only visual, but also auditory, hence the term *audio-score*. In previous work with the *De Caelis* ensemble over a period of ten years, [5] audio-scores have proved to convey an invaluable tool for the learning process of a piece of microtonal music. Audio-scores can also be very effective in performance since they simplify the task of the singers, free them from tuning forks and from the anxiety of getting lost in a difficult passage, both in terms of intonation or temporal coordination.

In this work however, audio-scores were always thought of as an extension of, rather than an alternative to notation. One of the greatest assets of musing-reading is that it allows the performer to preempt upcoming difficulties, whereas audio fluxes do not.

[3] http://animatednotation.com/composers.html

[4] The term of 'Computer-Aided Performance' was coined by M. Malt for the realisation of John Cage's *Concert for Piano and Orchestra* [5].

[5] The *De Caelis* ensemble has actively supported the project and commissioned five pieces using such systems.

1.2 Screen-Scores

In both experimental music and classical repertoire, tablets are increasingly replacing sheet music. However, the size of phones or tablet screens rarely matches that of large printed scores, and screen-based notation typically requires large font sizes and more page turns than its printed counterpart. A balance must therefore be found between displaying what is executed in the moment, and what is coming next, so as to always convey the most useful information to the performer. Indeed, problems may arise if a difficult passage (e.g. a high pitch or a long phrase) suddenly pops up at the start (at the left-end) of the page: breathing, or preparing an attack on a certain pitch, in sight-reading situations particularly, can require a few seconds of preparation that must be taken into account when creating the animated score.

In screen-based animated notation, the representation of time often contrasts with traditional sheet music. *SmartVox* for instance, uses notation realised in *bach.roll*, a Max object for proportional notation, with a cursor moving from left to right. This unfolding of time, inspired by digital audio workstations, offers an intuitive representation of speed and duration of musical events. Although it differs drastically from the classical "bars and beats" notation of rhythm, this solution proved to be very useful for singers since they need to anticipate the duration of a musical phrase to come in order to control their breath accordingly.

This proportional representation of time also profoundly simplifies temporal coordination between performers, particularly in non-pulsed music. For an inexperienced singer for instance, the main advantage of having his part displayed on a dedicated screen is the fact that he cannot get lost. Indeed, the screen score typically displays at any given moment only what is happening in his own part.

Such forms of augmented notation are therefore very convenient when singers have long parts to learn, when they are distanced from one another in the performance space (e.g. around the audience), or if the piece requires singing with microtonal intervals.

1.3 Early Prototypes

The *SmartVox* system was preceded by a series of prototypes with similar functionalities, developed over the past years.

Wired Cable Systems

The piece *Deserts* (2007) [6] used a computer to send audio-scores realized on a digital audio workstation. In this setup, each performer received two types of audio cues through an earpiece:

- A sequence of microtonal pitches.
- A series of clicks in an individual tempo.

Several techniques were investigated to distribute the individual audio cues to the performers, including mp3 players and wireless systems such as infra-red and high-frequency headphones as well as Bluetooth-based systems.

[6] Composed by J.Bell and commissioned by the *De Caelis* Ensemble

However, these systems were not robust enough to be deployed in concert performances so that wired systems, that usually consisted of a multichannel sound-card wired to headphones, were always preferred in such situations. These systems therefore locked the number of performers to the available output channels and impeded the ability to distribute the performers in the concert area.

Wireless Native Application

The piece *De Joye Interdict* (2014), also commissioned by the *De Caelis* Ensemble, used *iVideoShow*, a commercially available iOS application. The wireless setup was robust enough to guide performers in a concert situation. The application allowed the playback of individual video sequences on the iOS devices remotely controlled through Open Sound Control messages sent over a Wi-Fi network. This allowed the performers to be placed at a significant distance from each other. However, however *iVideoShow* presented several limitations:

- The system was difficult to improve and to deploy over cross-platform devices (iOS only).
- The synthesized audio cues, realized on a digital audio workstation, were difficult to control.
- The graphical notation captured as static screenshots from a notation software remained very close to traditional sheet music.

2. TECHNICAL SETUP / DESCRIPTION

The *SmartVox* application is based on the *Soundworks* framework [6]. The audiovisual scores – distributed as simple video files – are produced using the *Bach* (Bach Automated Composer's Helper) [7] environment in Max. [7]

2.1 Generating Audiovisual Scores Using the *Bach* Environment

The audiovisual scores used in the application are realized with *Bach*, a Max library for real-time computer-aided composition. In this environment (*bach.roll* or *bach.score*), each note of the score can be associated with metadata. Here, the feature is used to configure and control the synthesizer [8] that creates the audio cues, directly from the notational environment.

During the composition process, this particular setup allows for a workflow that consists in sculpting each vocal line with its appropriate pitches (in eighth-tones), text, intensity curves, formants, elocution velocity, glissandi, with a real-time audio feedback (see Fig. 2, 3).

The audio-score often only sounds when the performer should sing, yet some useful information can be provided to the performers during long silences: these audiovisual cues can deliver in advance the musical phrase that is coming next. In Figure 4, the lower stave is an anticipatory cue (the performer just listens), and the upper stave is an audio-guide (the performer sings along with the earpiece).

[7] https://cycling74.com/

[8] The *psych* module for Max - this module performs high quality pitch correction (auto-tune) and polyphonic harmonizing of monophonic audio sources.

Figure 2. A sample of spoken text stored inside the *filename* slot of each note (or group of notes) of the *bach.roll* object.

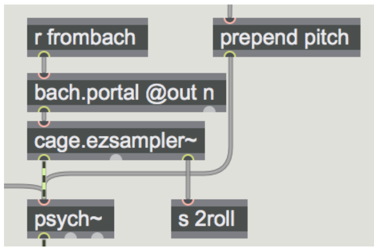


Figure 3. Excerpt from a *Max* patch showing the *cage.ezsampler* object retrieving metadata stored in the *Bach* score. The extracted information is then sent to the *psych* object, which transposes the given sample of spoken text to the defined microtonal pitch.

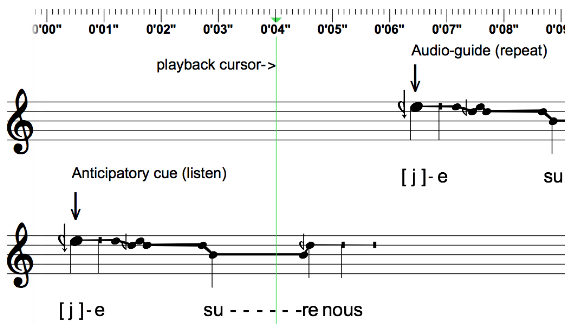


Figure 4. Screenshot of the video captured from the rendering of the *Bach* environment and played back by the mobile application based on the *Soundworks* framework.

2.2 Distributing Individual Scores to the Performers Using the *Soundworks* Framework

The system that distributes the audiovisual scores created with *Bach* is entirely based on web technologies, and more specifically on the *Soundworks* framework. [9] *Soundworks* provides a set of services – such as synchronization, network messages, distributed states, creation of groups of clients – that aims to solve problems common to distributed and synchronized web applications centered on multimedia rendering. The framework is written in *Javascript*, with a server side based on *NodeJS*. [10]

The *SmartVox* application consists of two web clients, the *player* and the *conductor*, that can be executed in any recent web browser on mobile devices (e.g. smartphones, tablets) and laptops. The real-time communication between

[9] *Soundworks* has been developed in the framework of the CoSiMa research project funded by the French National Research Agency (ANR) and coordinated by Ircam.

[10] https://nodejs.org/en

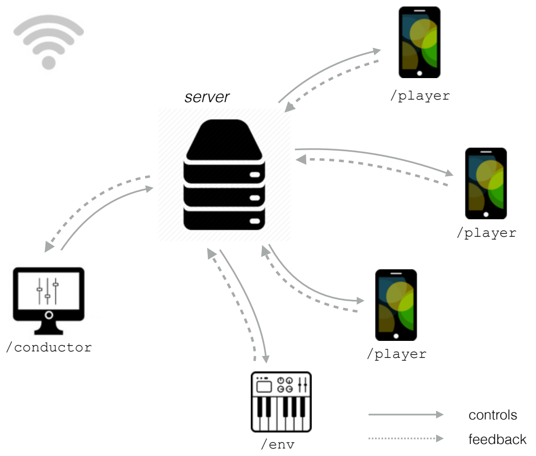


Figure 5. Architecture of the *SmartVox* application.

clients is achieved through the *WebSocket* protocol [11] (see Fig. 5).

*The **player** client*

This client, dedicated to the performers, is essentially a remotely controlled and synchronized video player.

Figure 6. Screenshot of the *player* client - drop-down menu showing the available parts of the score.

When entering the application, the performers are required to choose their part among the available ones (see Fig. 6). Once done, the corresponding audiovisual score is sent to the performers by the server. When the video is received on the device, further interactions with the score are locked in order to prevent the performers from accidentally changing the temporal position in the video, and thus ensure correct temporal coordination among all performers.

Additionally, this client can be used for the rendering of audio files (mp3) and/or videos (mp4) dedicated to the audience, through loud speakers and projectors.

*The **conductor** client*

The second client is dedicated to the choirmaster (see Fig. 7). Its role is to control the global and distributed state of the application. Through this interface, the conductor can therefore control the playback of the audiovisual scores:

- Start, pause and stop the video.

[11] https://www.w3.org/TR/websockets/

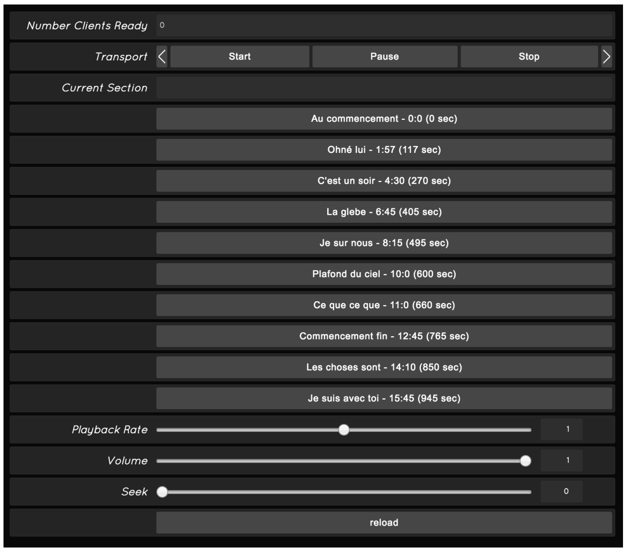


Figure 7. Screenshot of the *conductor* client.

- Jump to a labeled section of the piece or to a specific playback time.
- Change the playback rate (i.e., the speed) of the video without altering the pitch.
- Change the volume of all connected clients.

The interface also provides feedback (e.g., number of connected clients) to the choirmaster - information that proved to be of primary importance in concert situations.

Configuration

The application can be configured through a data structure that defines the path to the video files as well as the positions and labels of the sections of the piece. This allows for easily adapting the application to the content of a given piece:

```
const score = {
  duration: 20 * 60, // seconds

  // define the different parts
  parts: {
    'soprano-1': {
      file: 'videos/soprano-1.mp4',
    },
    'soprano-2': {
      file: 'videos/soprano-2.mp4',
    },
    // ...
  },

  // define the different sections
  sections: {
    alpha: {
      time: 0,
      label: 'First section',
    },
    beta: {
      time: 117,
      label: 'Second section',
    },
    // ...
  },
};
```

Both interfaces are dynamically generated according to this configuration, the list of sections in the *conductor* interface, and the list of available parts in the *player* interface.

2.3 Case Report: *Au commencement*

The application has been prototyped in parallel with the composition and rehearsal of the piece *Au Commencement* by the *Mangata* ensemble. One of the objectives of the project was to help the singers to tune-in (*i.e.* match spectrally) with the fixed-media electronics. The wireless and cross-platform system allowed singers to do so, while being placed around the audience (see Fig. 8).

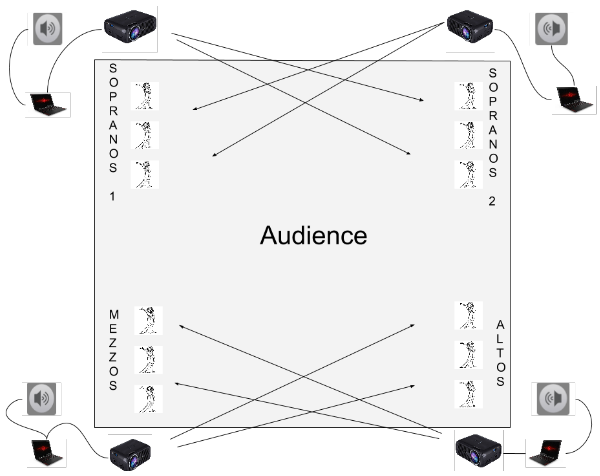


Figure 8. Spatial configuration of performers, lightings and loudspeakers.

A first public rehearsal of the piece revealed several issues with the user interface design. Indeed, the setup was already stable in rehearsals, but the challenges that arose in the public performance still required several modifications. Discussions with the choirmaster and the performers revealed the source of these problems: the most significant feedback concerned the lack of information about the current state of the system (in both *player*'s and *conductor*'s interfaces). As a result, several features were introduced in the next version of the application:

- On the *conductor* interface, display of the number of connected players, and addition of a button allowing to reinitialize all connected players.
- On the *player* interface, display of the name of the chosen part, and removal of the video controls.

The premiere of the piece took place in the *Notre Dame de Bon Secours* church in Paris [12]. Four groups of singers were situated at a significant distance from each other. Each of the twelve singers had a smartphone and earphones. Four additional computers, acting as generic *player* clients and connected to loudspeakers and projectors were placed in the four corners of the area (see Fig. 8). The technical setup confirmed the convenience and cheap cost of wireless cross-platform web technology: in spite of the disparity of the users' devices, communication was established quickly through the web browser, between Android phones, iPads, iPhones (the singer's devices), OSX and Windows (environmental videos).

[12] A recording of the premiere of the piece is available at: https://youtu.be/uVGPa1Z6Ji8

Acoustically, from an audience point of view, the extreme spatial separation between sources produced an immersive feeling and clarified the listening experience. This panoramic sonic image nevertheless let voices and electronics blend together successfully, thanks to the harmonic/spectral match between the two, and because of the natural reverberation of the church.

3. DISCUSSION

3.1 Pros

Technically, in its current state, the application demonstrates promising assets:

- Using web standards and Node.js, the application can be executed on a large range of platforms and devices.
- The architecture of the application allows for easily adapting it to different pieces by modifying its configuration.
- The application can be setup effortlessly and quickly in rehearsals and concerts using a laptop running the server and a Wi-Fi router.
- During rehearsals, the conductor client allows for flexibly navigating within a given piece.

From the performer's point of view, *SmartVox* is perceived as a useful device which combines acoustic and visual stimuli to help interpret challenging polyphonic scores, wether in rehearsals, performance, or pedagogical contexts. Also, experience proved that audiovisual scores surprise groups of performers in a positive way, which can be used as an impulse for challenging and imaginative musical/performative experiments.

Finally, from a compositional point of view, the piece *Au Commencement* demonstrated that the setup accelerates the learning process, facilitates the realisation of microtones, and allows to place performers at a large distance from one another, without putting them at risk in performance.

3.2 Cons

Feedback from the singers showed that audio cues sent through earpieces, while being useful in difficult passages, can tend to impede mutual listening. Further experiments will therefore propose visual cues only (or audio cues only), depending on the musical passage, in order to optimize the quality of the information given to the performer. [13]

Also, the clock synchronization between clients provided by the *Soundworks* framework [8] should be integrated in the application. Such precise time control among clients would thus allow for more complex rhythmical writing.

3.3 Use Cases

SmartVox owes its development to professional ensembles (*De Caelis* and *Mangata*) who tested the prototype in rehearsals and concerts, and gave invaluable feedback at each

[13] The exact delimitations of these cues could be defined iteratively from discussions with the singers and choirmaster together with an A/B testing strategy to speed up the process.

stage of its development. Since then, however, the application has also been used in different contexts:

Pedagogy

Recent experiments in conservatoires [14] demonstrated that, for children, a distributed mobile application can be evocative of a multiplayer game. This playful aspect helps to focus their attention on challenging music theory notions (*solfege*). In a pedagogical piece composed for this system, the notation purposefully conveyed the same pitch information in four different ways:

- Sound frequency: a synthetic voice sings on a given pitch, e.g., A = 440 Hz (audio).
- Spoken words: the synthetic voice pronounces the corresponding phoneme '*La*' (audio).
- Symbolic notation: the corresponding pitch is showed on the musical stave (visual).
- Written text: the phoneme '*La*' is written below the stave (visual).

After a few sessions, groups of ten to twenty children were able to sing in tune complex three-parts polyphonies.

For older students (undergraduates), *SmartVox* has been used to sing extracts of a motets of the Ars Nova, psalms of the Renaissance, as well as examples of early *solmisation*.

Amateur Choirs

A cappella choral singing requires competencies such as vocal skills, intonation and music reading. This often restricts ancient and contemporary repertoire to a small group of specialists. The audio and visual guides provided by *SmartVox* therefore seek to give accessibility and exposure to works otherwise judged too difficult (e.g., motets of the Renaissance or atonal music), for choirs of all levels.

Conclusion and Future Work

The present article described *SmartVox*, a web-based application specifically designed to help choristers in the realization of challenging pieces (i.e. including intricate performative aspects such as microtonality and unusual placement of performers in space). The application proved to be succesful in rehearsals, performances, as well as in pedagogical contexts.

Currently, the application is deployed over a private wireless local area network through Wi-Fi. The connected clients therefore require the physical presence of the server where the performance takes place. However, the application being completely implemented based on standard web technologies, it could be easily hosted on a public remote server and thus accessed over the Internet. While not suitable in concert situations, this feature will make an important difference in terms of spontaneous access and dissemination of the application, especially in pedagogical contexts.

[14] The recording of a *SmartKids* reading session is available at: https://youtu.be/hlHAeiWT28Y

In future versions, the application could also be expanded to allow for a better appropriation by the performer. The audiovisual score could then be used by the choirmaster during the rehearsal process, to make notations about the playback of the video (e.g. *accel.*, *fermata*, *crescendo*...). This would then be communicated to the performers.

The positive results thus far encourage us to believe that *SmartVox* will continue to be an innovative and useful form of musical notation.

Acknowledgments

We are deeply grateful to the following people for their precious support and advice: Andrea Agostini, Julian Anderson, Richard Baker, Eric Daubresse, Stefano Gervasoni, Daniele Ghisi, Marisa Gupta, Bryn Harrison, Gregoire Lorieux, Paul Newland, Hector Parra, Nye Perry, Norbert Schnell, and James Weeks.

References

[1] E. Ghent, "Programmed Signals to Performers: A New Compositional Resource," *Perspectives of New Music*, vol. 6, no. 1, pp. 96–106, 1967.

[2] P. Kocher, "Polytempo Network: A System for Technology-Assisted Conducting," Institute for Computer Music and Sound Technology, Zurich University of the Arts, 2014.

[3] G. Daunoraviciene, "Sound architecture of Rytis Mažulis' microstructural canons (from 100 to the 3,448275862 cents)," *Menotyra*, vol. 30, no. 1, 2003.

[4] J. Bell, "Audio-Scores, a Resource for Composition and Computer-Aided Performance," Composition Doctorate, Guildhall School of Music and Drama, 2016.

[5] B. Sluchin and M. Malt, "Interpretation and Computer Assistance in John Cage's Concert for Piano and Orchestra (1957-1958)," in *7th Sound and Music Computing Conference Proceedings*, 2010.

[6] S. Robaszkiewicz and N. Schnell, "Soundworks – A playground for artists and developers to create collaborative mobile web performances," *http://architexte.ircam.fr/*, 2015.

[7] A. Agostini and D. Ghisi, "Real-time Computer-Aided Composition with BACH," *Contemporary Music Review*, vol. 32, no. 1, April 2013.

[8] J. P. Lambert, S. Robaszkiewicz, and N. Schnell, "Synchronisation for Distributed Audio Rendering over Heterogeneous Devices, in HTML5," in *Proceedings of the 2nd Web Audio Conference (WAC-2016)*, 2016, pp. 6–11.

NOTATED CONTROL AS COMPOSED LIVENESS IN WORKS FOR DIGITALLY EXTENDED VOICE

Kristina Warren
University of Virginia
kmw4px@virginia.edu

ABSTRACT

This study argues that learning of varying control mappings in digitally extended voice works imbues body and memory into liveness. First, the author's extended voice practice is discussed. The Abacus, a unique, microphone-mounted, Arduino Teensy-based musical interface, controls granulation of live vocal samples. There are sixteen pre-composed mappings of Abacus control data (eight toggle switches) to granulation parameters, and mapping changes regularly. An animated screen score provides manual toggle control instructions, which didactically supply information on current mapping. Subsequently, discussion turns to works by other extended voice practitioners and to a larger context of screen scores and musical games. Building outward from notions of vocal intimacy and presence, extended voice uses technology for temporal exploration of timbre. Screen scores and musical games highlight learning, but typically utilize an unchanging control mapping throughout the piece or game. My work constitutes a novel intersection of these practices, arguing that repeated, notation-driven learning of the action-sound relationship thematizes complex interactions between body, temporality, memory, and presence.

1. INTRODUCTION

Voice is a complex phenomenon, but embodied presence is usually considered a crucial part [7, 17, 19]. Recent scientific-musical studies of voice begin to address connections between embodiment and liveness, but frequent division of labor between technologists and vocalists hinders integration of performative and technological methodologies. By contrast, extended voice practitioners, including Andrea Pensado, Ami Yoshida, and the author, synthesize technological, compositional, and improvisational methodologies and thus directly address liveness during electroacoustic performance.

This study discusses two of the author's recent compositions for extended voice: "couldn't" (2016) and "for ami" (2016).[1] First, the extended voice setup is discussed in the context of "couldn't." Subsequently, I discuss the use of a screen score of control instructions as a method of composing liveness in "for ami" (2016). Finally, these compositions are related to other extended voice works and to larger discussions of screen scores and musical games. I argue that screen score works for extended voice create a novel form of liveness by employing the liveness-through-didacticism of screen scores and by questioning the connotations of intimacy, embodiment, and presence typically ascribed to voice.

[1] Excerpts at http://kmwarren.org/tenor17.html

2. VOICE, ABACUS, MAX/MSP

2.1 Live Vocalization

My vocalization in my voice-electronics compositions includes normative singing, occasional speech, and frequent extended vocal techniques. Typically, the voice provides fodder for the granulation system and tries to achieve a dialogue with the electronic sounds, but feature extraction can impart structural importance to particular vocal sounds as well. For instance, "couldn't" uses undertone singing, or vibration of the false vocal folds at one-half or sometimes one-third of the fundamental frequency, to trigger changes between the two patch states. Feature extraction in MaxMSP identifies undertone singing through its distinctive ratio (2.1-2.6) of spectral centroid to fundamental frequency.

Pitched	
Harmonics	slow alternation [u] (har. 1-2 above)
Undertone sing	false vocal folds (har. 1-2 below)
Ululation	fast chest-head register alternation
Inhale sing	pure / noisy (resembles bitcrush)
Lip squeak	upper teeth, moist lower lip, inhale
Pursed lips	pitched squeak, (sub)audio rate
Pressed squeak	high in vocal tract / post-nasal
Unpitched	
Duck call	air against mid-back hard palate
Glottal stop	beginning / end of note
Lip buzz	usually sub-audio rate / bursts
Press / fry	bursts, can merge into audio rate
Epiglottal click	in- / egressive, single or sequence

Table 1. Extended vocal techniques (selected)

Many extended vocal techniques are very quiet. These non-normative vocal sounds frequently originate from non-larygneal oscillators such as lips with teeth, air against the hard palate, or saliva against cheeks. My work increasingly tries to attend to the mouth as origin of these tiny, detailed sounds. The novel control interface called the Abacus is mounted directly on the microphone clip; the necessity of bringing the hand near the mouth in order to control draws audience attention toward the mouth and its small sounds. "Couldn't" and earlier works include dense electronics, while the voice purposefully struggles to co-exist by imitating the electronics' rhythm or timbre. While this voice-electronics competition is evocative, it is not ideal for drawing attention to small, mouth-originating sounds. Thus, "for ami" and more recent works use primarily less-dense electronic textures and aim for a gentler or more subtle sound world.

2.2 Abacus

The Abacus is a novel interface used for controlling digital processing of voice. It consists of an Arduino Teensy, which sends Serial information via USB connection, along with eight toggles, four potentiometers, one button, and one LED. Thermoplastic affixes the Abacus to a Shure SM-58 microphone clip. Crucially, this interface permits simultaneous vocalization and control.

Figure 1. Abacus interface

<table>
<tr><th>Toggle 1</th><th>Toggle 2</th><th>Toggle 3</th><th>Toggle 4</th></tr>
<tr><td>listen for undertone singing</td><td>solo one processed voice</td><td>live voice: random processing</td><td>record live voice/ processed sound</td></tr>
<tr><td>Toggle 5 (voice 1)</td><td>Toggle 6 (voice 2)</td><td>Toggle 7 (voice 3)</td><td>Toggle 8 (voice 4)</td></tr>
<tr><td colspan="4">State 1 granular/ not | State 2 follow/ ignore rhythm</td></tr>
</table>

Table 2. Abacus, control mapping in "couldn't"

The streamlined design of the Abacus is conducive to varied control mappings. Within my piece "couldn't," the mapping of Toggles 5-8 varies with patch state; in one state they control timbre, and in another state, rhythm and density.

An early control mapping scheme, developed in 2015-2016, is much more parametric. In this mapping scheme, Rhythm, Pitch, and Noise are three control spaces which route toggle pairs 3-4, 5-6, and 7-8 to distinct sub-parameters such as Meter, Interval, and Timbre. Extended vocal techniques, identified through feature extraction, trigger navigation among the three overarching control spaces [16]. This mapping yielded musically dissatisfying results; complexity prohibited control fluency and expressivity. This led to ongoing research into streamlining control mapping, as in "couldn't," while still varying mappings during the composition, attempted initially in "couldn't" and more meaningfully in "for ami."

2.3 Real-Time Processing in MaxMSP

Real-time processing of live vocal samples occurs in MaxMSP. Several types of processing occur: rhythmic granulation, pulsar granulation (i.e., streaming an alternation of grain and silence [11]), and wavetable synthesis.

Processing of the live signal, rather than live samples, is also possible, using techniques such as transposition, distortion, and delay. The patch is modular, such that any signal (voice, granulation, or wavetable synthesis) can be heard 'clean' or sent to live processing and/or subsequent granulation. The overall aesthetic goal is to explore similarities between vocal and digital sound worlds, particularly the shared noise potentials of each.

My piece "couldn't" is fairly rigid in its implementation of pulsar granulation. A few possible values exist for grain start time and length, and wait time between grains is always 10 ms. Slight, probabilistic jitter in grain start time, length, and playback speed, calculated independently for every grain, provides some musical interest. Nonetheless, this rigidity of pulsar granulation settings contributed to a desire for more flexibility in later works.

<table>
<tr><th colspan="2">Start time (0 - 4000 ms)</th><th>Len (ms)</th><th>Wait (ms)</th><th>ST, len jitter</th><th>Speed jitter</th></tr>
<tr><td>400</td><td>1800</td><td>30</td><td>10</td><td rowspan="4">10% chance +- 3% ST and/or len</td><td rowspan="4">67% chance +- 2% speed</td></tr>
<tr><td>1000</td><td>2100</td><td>80</td><td></td></tr>
<tr><td>1500</td><td>2200</td><td>100</td><td></td></tr>
<tr><td>1700</td><td>2800</td><td>140</td><td></td></tr>
</table>

Table 3. Pulsar granulation parameters, "couldn't"

3. NOTATED CONTROL

My recent piece "for ami" employs vocalization, Abacus, and live sample granulation in MaxMSP, with frequent variations in the Abacus control mapping. The Abacus makes it easy to vocalize and control at once, so the primary goal in this piece was to go a step further and attempt to learn the changing control mapping in order to explore liveness.

"For ami" is nine minutes long and consists of a series of miniatures (durations 0:15-2:00) separated by pauses. Each miniature represents a different control mapping. Some miniatures have a pre-composed duration, and others have a free duration such that I trigger the end of the miniature ad lib. The piece is inspired by and named for extended voice practitioner Ami Yoshida. Her album *Tiger Thrush*, consisting of 99 untitled miniatures, is a meditative, inspiring exploration of vocal and electronic timbre.

3.1 Mappings

There are sixteen pre-composed control mappings. Each allows control of some parameters of pulsar granulation, and assigns other parameters uncontrolled values (either constant, or simple LFO-varied). Most mappings do not map control data directly or uniformly to parameter values, but rather to LFO parameters for regular temporal variation of granulation parameters. This sculpting of complex LFOs yields fine control of timbre. In addition, mappings incorporate variable gain gating on the granulated signal, yielding a more ‘human’ sound which includes rests. Finally, each mapping also includes some associated toggle instructions for the screen score.

(values in ms unless otherwise noted)				
	tog 1-2	**tog 3-4**	**tog 5-6**	**tog 7-8**
1	live vox, pan alt. = 0.5, 1, 4, 10 Hz	ST = 400, 1600, 2800, 3800	len = 3, 10, 25, 41	wait = 5, 30, 70, 90
2	live vox, # transposed versions = 0, 1, 2, 3; # cents = -75, -20, 35, 60		ST = 200, 320, 3520, 2640	len = 2, 8, 12, 18
3	alternate ST_1,2		len, alt. (5-12) time = 160, 600, 980, 2300	wait, alt. (20-30) time = 140, 550, 900, 2000
	ST_2 = 0, 480, 1200, 2000	alt. time = 80, 300, 500, 1200		
4	len, wait LFO		ring mod. on LFO	
	rate = 3, 20, 47, 80 Hz	depth = 1, 10, 50, 100 Hz	rate = 5, 8, 10, 20 Hz	depth = 0.01, 0.1, 1, 4 Hz
5	ST, len LFO (alt. slow-fast)			alt. long-short wait, duty cyc = 0.6, 0.8, 0.9, 0.95
	duty cycle = 0.5, 0.6, 0.7, 0.9	rate_fast = 1.9, 5.4, 11, 36 Hz	depth = 2, 10, 20, 40 Hz	
6	4/4, 48 bpm (mute, un-mute alternating beats)			
	beat 4 ST = 800, 1600, 2400, 3600	beats 1,3 len = 1, 4, 8, 16	beats 2,4 len = 3, 5, 10, 30	b. 2,4 time til mute = 10, 50, 100, 300
7	4/4, 132 bpm			
	beat 1 ST = 0, 400, 600, 800	beat 2 ST = 1200, 1400, 1600, 2000	beat 4 ST = 3200, 3400, 3680, 3960	beat 3 wait_short = 5, 10, 18, 36
8	4/4, 21.6 bpm (muted, occasional 16ths sound)			
	alt. ST_1,2 every 40 ms		len_1,2,3 (alt. every 152 ms) = 1-2-3, 2-4-6, 4-8-12, 8-16-24	wait_2 (alt every 600 ms) = 10, 20, 50, 100
	ST_1 = 40, 360, 560, 720	ST_2 = 800, 960, 2440, 2720		

Table 4. Control mappings 1-8, "for ami"

In all mappings, the eight toggles are treated as four pairs, each outputting values 0-3. Mappings 1-2 scale toggle-pair values more or less directly to values of live vocal signal processing and grain start time, length, and wait. In the other mappings, however, toggle-pair values contribute mostly to time-based variance of parameters, for instance through low frequency oscillators (LFOs) or rhythmically metered variance.

This routing of control data to rate of change rather than to discrete parameter values comprises a second-order treatment of control data. This yields a more complex and musically pleasing sound. Often, processing changes are only discernible after several seconds of listening. This is a purposeful compositional decision which slows the performance pace, lends a meditative quality, and creates time for purposeful decision-making about exact content and timing of vocalization.

On loading the patch, a random order of the sixteen mappings is generated, including possible nonconsecutive repetition. The unpredictable order of the sixteen mappings renders them surprising enough to require re-learning during performance.

3.2 Screen Score

An animated screen score presents real-time instructions for manual operation of the Abacus toggles. These instructions are didactic in nature, uniting with the audio result to progressively reveal information about the current mapping. The large central toggles report current state and show toggle instructions, with a red-yellow-green countdown to enable performer preparation. The smaller toggles in the upper right corner provide redundant information, namely the summation of a few recent and upcoming toggle steps. This animated score is viewed on a laptop screen during performance and, if circumstances permit, may also be projected onto a larger screen visible to the audience.

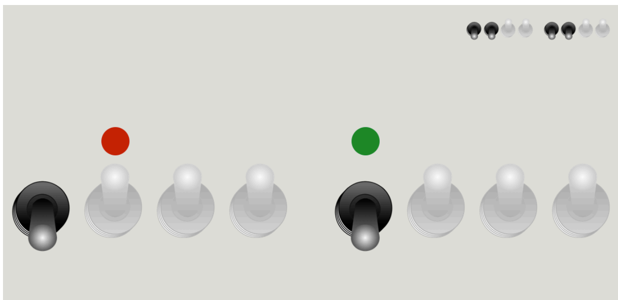


Figure 2. Video score screen capture, "for ami"

In combination with the audio output, the screen score serves a didactic function. Each mapping carries associated video instructions which indicate the content of the mapping. For instance, in mapping 7, the tempo and meter are rapidly audible. The first instructions might require toggles 1-2 = 0 then 2 (as binary numbers, separated only by one flip of toggle 1) to illustrate the sonic difference on beat 1. Next, similarly, the instruction might require toggles 3-4 = 0 then 2, demonstrating the sonic change on beat 2.

Some randomization of instructions occurs, so that even miniatures using the same mapping are distinct.

Recording of live samples also complicates matters. There are up to twenty distinct granulation voices which may be processing different buffers, in any combination. Typically, I record several new buffers during the piece, where audio source can be either the vocal signal or, cyclically, the granulation output.

To clarify, the screen score only notates manual control gestures. Vocal material, including techniques shown in Table 1, is fully improvised. I do not enjoy realizing vocal notation because I feel that accuracy takes precedence over expressivity. Thus, “for ami” notates control gestures and utilizes vocal improvisation.

4. COMPOSED LIVENESS

Liveness in “for ami” is premised on several key assumptions. First, Abacus-based control gives a greater sense of presence in the moment than would a non-interactive system. Second, varying the control mapping several times during the piece constitutes an opportunity for performer learning, which further enhances liveness. Third, as in many musical systems and idioms, some improvised deviation from the score is possible once the rules are well enough understood, comprising yet another layer of liveness.

It is widely acknowledged that digital interfaces are uniquely powerful because they allow myriad relationships between physical action and sound. “For ami” acts directly on this possibility space. Cyclical learning – starting afresh with each miniature – foregrounds liveness, and learning is incentivized by the possibility of improvisation with the learned mapping.

Extended voice is unique in treating memory as a part of liveness. Granulation of live vocal samples, for instance, is intended partly to confuse listeners’ short-term memory of vocal utterance. A noisy sound which initially seems very remote from voice may, through variation in granulation parameters, be gradually revealed to be vocal in origin. In other words, timbral evolution makes electronic timbres present, while the acoustic voice, which traditionally connotes intimacy and presence, becomes a thing of the past.

Timbral similarities between voice and electronics emerge despite the time-lag implicit in live sample-based work. Unlike parallel processing of voice, for instance vocal pitch-following to yield fundamental frequency of an oscillator, live samples take a few seconds to record, and then several more seconds to process using manual control. In other words, liveness in extended voice is a dialogue with a digitized version of oneself from several seconds ago, and control of this digitization may in turn influence the live vocalization.

5. DISCUSSION

This section addresses Ami Yoshida’s album *Tiger Thrush* (2003) [18], which inspired the author’s composition “for ami,” and Andrea Pensado’s album *Without Knowing Why* (2015) [9]. Yoshida varies pauses and durations within and between miniatures to create novel temporal interactions between extended vocal techniques and electronics, while Pensado’s work is unique among extended voice for its rapid variation among noise timbres. Thus, both Yoshida and Pensado dually explore temporality and timbre. Similarly, extended voice at large links timbre and temporality by including memory within performative liveness. The voice has the potential to be digitized at any moment, making it an instrument of the past, while electronics indicate the present.

This section will also address studies of screen scores and video games. Whereas these two media aim for control consistency and transparency, extended voice employs varying control mapping, which unsettles intuition and renders the body strange, just as voice is rendered strange through timbral exploration and digital processing.

5.1 Yoshida, *Tiger Thrush*

Ami Yoshida’s album *Tiger Thrush* consists of 99 untitled miniatures. Yoshida’s self-defined style of “howling voice,” or quiet speaking and screeching sounds often produced through inhalation or high-pressure exhalation, joins found objects and environmental background sounds in this album.

Tiger Thrush manifests the timbral-temporal exploration common in extended voice. Some tracks, such as #8, are single vocal utterances of only a few seconds’ duration. Though such tracks initially seem to function as non-structural palate cleansers, their presence throughout the album suggests that something more is at play. By contrast, other tracks, such as #5, loop a single vocal phrase, either mono- or polyphonically, for several minutes. At first blush, these longer tracks seem important formal anchors within the album. Nonetheless, the stillness and persistence of *Tiger Thrush* gradually suggest an inversion: micro-tracks come to read as intense sound bytes which eschew embellishment, while longer looped tracks provide a sense of familiarity and pad the more concise vocal statements.

Yoshida’s temporal exploration of timbre is also apparent in the non-vocal tracks, of which #45 is a prime example. This track, consisting of looped, high-frequency background noise, follows shortly on the heels of the previous track and lasts almost five minutes. Track #45 utilizes the same room noise present in Track #44, which contains several substantial pauses, but articulates a new musical statement because of its length and subtle variation of loops. Through slight manual adjustments to the time bounds of the loop, the listener begins to question whether the sound is changing or if this is in fact a perceptual mirage. Yoshida’s intermingling of silence and continuity serves to make timbre strange and to unsettle short-term sonic memory.

Delicate reverb and some reversed sounds subtly remind listeners of the presence of technology. Unlike many extended voice practitioners, Yoshida is minimal in her demonstration of technological control. The subtle compositional decision of whether to conclude a track partway through or at the end of a loop is one point of engagement with the technology. Other tracks underscore the sonic role of the technology itself. For instance, in Track #49 Yoshida performs a melody consisting of distorted microphone pops.

5.2 Pensado, *Without Knowing Why*

In her 2015 album *Without Knowing Why* and her recent live performances, Andrea Pensado explores a variety of voice-noise interactions. The titular character in “Rondo con Andreita” is likely the doll with which Pensado sometimes performs (e.g., Back Alley Theater, Washington DC, Sep 2014[2]) and which is mentioned in the album liner notes. “Rondo” and the Back Alley performance are distinct versions of the same material, where the A section of the rondo form consists of quiet speech, little processed, while the alternating contrasting sections are much noisier. These noise sections are characterized by rapid timbral changes (rate of change = 2-4 Hz). The noise initially follows the melodic and/or rhythmic contour of Pensado’s voice but in later non-A sections grows more independent and increasingly masks the voice.

The humor and weirdness of Andrea Pensado’s use of, apparently, a doll named “little Andrea” augment the questions of control raised by the work. The voice can be used for both expression and control, and these functions sometimes become indistinguishable in Pensado’s sound. The apparent spectral following which drives synthesis is evidence of hands-free vocal-control work. Yet Pensado also uses her hands to animate the doll as though it is speaking or listening, and these motions seem to trigger sonic changes. The doll may contribute to the sense of hand-sculpted rhythmic detail, or this audio-visual link may be completely imagined. The rapidity of timbral changes further obscures control source.

Notably, although harsh noise timbres prevail, compositionally-trained phrasing is also apparent. Gestures are not random, but rather occur within some structure. For instance, the garbled, chorused speaking voice beginning at 3:07 in the left channel uses quantized, almost vocoded, frequencies emphasizing a range of about a minor sixth. Noise sections do not begin abruptly, but are instead prepared by brief, growing interruptions often panned centrally.

Pensado’s frequent changes in electronic timbre are unique in the landscape of extended voice. This protean quality may arise in part from deep familiarity with the capabilities of her technology. Though Pensado identifies as an improviser who takes a “highly intuitive” approach to "using Max as her main programming tool” (from Pensado’s professional bio), her utterances nevertheless seem tailored to what the patch does well, for instance speech with larger than normal frequency range to provide interesting fodder for noise synthesis.

[2] https://www.youtube.com/watch?v=KspVGrJrhpg

5.3 Screen Score Works and Musical Games

Screen scoring is valued for its unique musical possibilities, including audience interaction [2], novel methods for representing time and texture [4,13], and conduciveness to formal re-combination and material generativity [15]. Screen scores innovate in musical expressivity, but, like traditional notated music of the Western canon, they often make certain assumptions about time and liveness, or lack direct connections to physical action. For instance, Kim-Boyle praises screen scores’ performative and compositional metaphor of navigation along a pathway [6], but this seems to rely on assumptions of time as a simple forward flow.

Musical games take a related approach. Pichlmair and Kayali’s taxonomy of musical video games includes two overarching types, Rhythm and Instrument, and seven important features, including Active Scores, Synesthesia, and Play-as-Performance [10]. Game-salient types of learning, such as “just in time” learning, are a novel synthesis of past, present, and future: instructional content, tailored to past experience, is delivered to aid completion of a near-future task [3]. Similarly, game pieces by composer-researchers treat games as toys; by learning the physical rules, or affordances, of these toys, players can achieve a flow state of creative interaction [5, 14].

Though screen scores and video games may employ different objects – acoustic instruments versus game controllers – both emphasize unchanging action-screen relationships throughout the piece or game. For instance, “natural mapping” of video game control, i.e., similarity between control action and animated game result, aims to create a smooth progression through states of learning, improvement, and expertise within the single control mapping [12]. On the rare occasion when control mapping does vary, for instance variation in control degrees of freedom to allow fluid motion through game space, this is treated as ‘under the hood’ algorithmic information unnecessary to the performer [8].

In contrast to the control transparency intended in screen scores and video games, extended voice is motivated by a desire to make the body strange. This exists first in exploration of vocal timbre through electronic extension of the voice, but also in varying control mapping. Many extended voice practitioners use minimal electronic setups such as laptop with one controller, where digital patch states govern variation in control mapping. Thus, the same physical action – pressing a particular button or flipping a particular toggle switch – could yield very different sonic results at different moments in the performance. This counteracts embodied intuition. The body is decentralized and becomes a thing of the past, and focus goes instead to digitized sound in the present.

Notated control is a form of choreography intended to inspire novel performance temporalities. In the context of my interface the Abacus, this choreography is seemingly minute in that it primarily addresses the fingers, but it has much broader ramifications. Bringing the hand close to the mouth draws attention to the mouth, an often overlooked musical site (distinct from the larynx). This renewed attention to mouth serves to thematize the complex interactions of embodiment and temporality. The frequent and repetitive vocal digitization in extended voice practice places body in the near past. The body is an input to, rather than an acting agent within, the present moment.

6. FUTURE DIRECTIONS

Extended voice practitioners explore novel forms of liveness and performance temporality, often through composed but improvisation-conducive technical methods, e.g., varying digital patch state to re-map control information. This is particular to extended voice because practitioners work within and against traditional assumptions that voice is intimate and present. They undertake prominently temporal variation of vocal timbre and explore voice-electronics as novel instrument.

Ongoing performance work will clarify the interactions between vocal expressivity, digital noise, and non-verbal communicative acts. My vocal work rarely uses text, so text-based theoretical precedents, including Barthes' notion of the 'grain of the voice,' are not directly relevant. Instead, I build upon physical vocal research which treats the voice itself as technology which is malleable for creative purposes [1].

Further research is needed into intersections between extended voice, screen scores, and musical games. Extended voice offers a novel response to notions of embodiment and presence, but research is needed regarding the responses of other musicians or of gamers to control mapping variation. By contrast, screen scores and game pieces offer the advantage of expertise through repetitive learning. While these media yield exciting possibilities of musical expression, continued study is needed into the contributions of complex forms of embodiment and liveness.

7. REFERENCES

[1] D. Z. Borch, J. Sundberg, P.-Å. Lindestad, and M. Thalén, "Vocal fold vibration and voice source aperiodicity in 'dist' tones: a study of a timbral ornament in rock singing," *Logoped Phoniatr Vocol*, vol. 29, pp. 147-153, 2004.

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

[3] A. C. Y. Hung, *The Work of Play*, Peter Lang Publishing, 2011.

[4] D. Jackowski, F. Melendez, A. Bauer, P. Hendrich, and C. Duchnowski, "Computer Game Piece: Exploring Video Games as a Means for Controlled Improvisation," *Proc. Int. Computer Music Conference 2014*, Athens, 2014, pp. 88-92.

[5] D. Kanaga, "Intro to Ludic Ecologonomy (Pt. 1)," *wombflash forest* [online journal], Apr 2015, http://wombflashforest.blogspot.co.uk/. Accessed 11 Nov 2016.

[6] D. Kim-Boyle, "Visual Design of Real-Time Screen Scores," *Organised Sound*, vol. 19, no. 3, pp. 286-294, 2014.

[7] C. Lane, "Voices from the Past," *Organised Sound*, vol. 11, no. 1, pp. 3-11, 2006.

[8] J. Laszlo, M. van de Panne, and E. Fiume, "Interactive Control for Physically-Based Animation," *Proc. Conference on Computer Graphics and Interactive Techniques*, New Orleans, 2000, pp. 201-208.

[9] A. Pensado, *Without Knowing Why*, FTR203, LP, 2015.

[10] M. Pichlmair and F. Kayali, "Levels of Sound: On the Principles of Interactivity in Music Video Games," *Proc. Digital Games Research Association Conference*, Tokyo, 2007, pp. 424-430.

[11] C. Roads, *Microsound*. MIT Press, 2001.

[12] P. Skalski, R. Tamborini, A. Shelton, M. Buncher, and P. Lindmark, "Mapping the road to fun: Natural video game controllers, presence, and game enjoyment," *New Media & Society*, vol. 13, no. 2, pp. 224-242, 2011.

[13] R. R. Smith, "[Study No. 50] [Notational Becoming] [Speculations]," *Proc. Int. Conference on Technologies for Music Notation and Representation*, Cambridge, 2016, pp. 98-104.

[14] P. Turowski, "Digital Game as Musical Notation," PhD dissertation, University of Virginia, 2016.

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

[16] K. Warren, "Composing and Performing Digital Voice Using Microphone-Centric Gesture and Control Data," in *Proc. Int. Computer Music Conference 2016*, Utrecht, 2016, pp. 439-442.

[17] A. L. Woloshyn, "The Recorded Voice and the Mediated Body in Contemporary Canadian Electroacoustic Music," PhD dissertation, University of Toronto, 2012.

[18] A. Yoshida, *Tiger Thrush*, IMJ-504, CD, 2003.

[19] M. Young, "Latent body – plastic, malleable, inscribed: The human voice, the body and the sound of its transformation through technology," *Contemporary Music Review*, vol. 25, no. 1-2, pp. 81-92, 2006.

NOTATING ELECTROACOUSTIC MUSIC FOR PERFORMERS FROM A PRACTITIONER'S EXPERIENCE

Terri Hron
Wesleyan University
thron@wesleyan.edu

ABSTRACT

This paper discusses notation practices and experiments within the electroacoustic performance and composition practice of the author. These spring from a performer- and performance-oriented position towards notation in a field that has traditionally catered more to notation for analysis and description. As such, the works and experiences discussed offer hybrid solutions and multiple formats to satisfy specific needs for the effective rehearsal and performance of electroacoustic music. The adaptation of tools specific to electroacoustic practice for more contemporary classical performers is discussed using examples from works written for and by the author in collaboration with other performers and composers.

.

1. INTRODUCTION

This short paper presents a field report of my experiences and experiments in the notation of electroacoustic music both as a performer and composer. As such, it is more of an artist statement than a description of methodology and results. In the last ten years, I have commissioned fifteen new works for recorder and electronics,[1] co-composed large-scale works for the same with Monty Adkins and Hildegard Westerkamp and written a dozen works for other performers/instruments and electronics. In the course of these collaborative experiences, notation has always been a puzzle, which I approach in a very practical way, with the performer's experience and needs very much at the forefront of my concern. My relationship to the score is predicated by my first vocation as a performer of early music, deciphering scores where clearly so much information about performance practice, aesthetics and poetics is not available. This affected my own notational practice in two ways: (a) an attachment to a performance score as the simplest possible mnemonic device to jog pre-existing knowledge of style, affect, technique, etc. and (b) a desire to create/have access to a repository of all that pre-existing, accumulated knowledge and detail about a work as part of notational practice. Clearly these are familiar issues that in my case are coloured by involvement with electroacoustic practice, whose multiplicity of technology evades normalization in notation. My approach is also very much affected by the blurring of roles and duties between composer and performer. The first part of this paper, then, will focus on multiple score formats in the transmission of electroacoustic music for specific instrumentalists and the second will discuss more recent scores I created as a specific performer within co-composition. My goal is not to propose any conclusive method, but to present some solutions drawn from a multi-faceted practice that has grappled with notation within electroacoustic music for some time.

2. MULTIPLE SCORE FORMATS

I began my musical life as an early music performer at a time when historical performance practice still largely pledged allegiance to the Romantic notion of *werktreue.*[1] Over time, I came to realize, however, that the only road to discovering how to successfully decipher early scores and whatever 'intention' might lie behind them was through performance knowledge and experience with *playing* original instruments. Though this performer's perspective and background seems to me fundamental, as Lydia Goehr and others have made abundantly clear the Romantic ideal – carried over and magnified in Modernism – has also necessarily permeated my understanding of a contemporary score's function as carrier of intentionality. I feel, however, that there is a twist: since my work – and that of so many working in electroacoustic music – is the result of collaboration, perhaps a score can carry a multiplicity of intentions. And perhaps it can also remain what it was in the very beginning: not the closest thing to the "work itself" but the simplest possible mnemonic device. In trying to reconcile these divergent aesthetics and needs, I started to experiment with the idea of multiple score formats: one to carry intentions, another to use practically in performance, and a third to satisfy the simplest practice needs. A hyperscore, a videoscore and a paperscore.

[1] Composers include: Jim Aliteri, WL Altman, Daniel Blake, Ronald Boersen, Juan Parra Cancino, Jorrit Dijkstra, Peter Hannan, Jenny Olivia Johnson, Emilie LeBel, Paula Matthusen, Darren Miller, Robert Normandeau, Laurie Radford, Elliott Sharp, Peter Swendsen.

2.1 Hyperscores

In the context of my doctoral thesis, *Composing idiomatically for specific performers: collaboration in the creation of electroacoustic music* [2], I wrote a piece for solo piano and electronics in collaboration with three pianists. My goal was to examine whether each performer would lead me to write differently for the same instrument, based on their particular interests, abilities and proclivities. When it came time to notate the work entitled *Maly velky Svet* (2013), I searched for a medium that would allow me both to document some of these differences in extra-musical form and to acknowledge the creative contribution these specific performers had made to the process. Basically, I was looking for a way to house the performance practice of the work, which had started with the first ideas and sketches, within the score itself.

Over time, a few other considerations came to light. The first was that a performance-practice repository within the score could be open-ended, in the sense of Eco's *open work* [3]. A hypertext document, where links could be added as comments without necessarily altering the source could encourage future performers to complete the work after their own fashion. Following from this after a fashion, a second consideration was a vision of the work's provisionality, in William Kentridge's definition: "There is not a script or a storyboard. There is a contingency to meaning and what can be gleaned from fragments coming together. A construction rather than a discovery. As with a drawing, a meeting of the world halfway. Only in retrospect does anything have determined inevitability" [4]. The choice of hypertext is a commitment to the non-linear experience of learning a musical work. It has always struck me that the end result of the activities of a musical interpreter is a temporally-defined performance with a beginning and an end, yet its preparation is a series of rabbit-holes and practice loops. A hyperscore encourages and recognizes this contemplative, out-of-time relationship to the score. Finally, I realized that *Maly velky Svet*, a work whose poetic side revolves around childhood and games (with musical references to other pedagogical/children's musical works such as Bartók's *For Children* and Schumann's *Waldszenen*), and that is technically not forbidding, could be a good introductory work for young players. If it were to fulfill such a role, it would have to be explicit and explanatory.

So, practically, what kind of information can be housed in such a hypertext version of the score? To give examples, I will refer to the second and third part of *Maly velky Svet*, entitled FATE and KNOCKING [5]. A hyperscore can include everything from biographical information about the collaborating artists (click Katherine in FATE or Luciane in KNOCKING), inspirational, poetic/aesthetic explanations about the movements (click on the titles FATE or KNOCKING), specific information about sounds and events in the fixed media parts of the electronics (click on F in FATE and B or highlighted section in D in KNOCKING), or explanations of specific techniques (highlighted sections of B and F in FATE). All of this information is of interest and is thus easily available to performers yet does not need to appear on the page during practice or performance. In the past, such information often resided in the preface or legend to a score, yet those predicate a linear experience of such information and seldom include video or audio material pointing specifically to certain events.

2.3 Videoscores

One of my ongoing concerns as a composer of electroacoustic music is providing performers with tools that allow them to rehearse with the electronics as often as possible. The reality is that most performers do not have access to studio monitors or a PA system that would allow them to practice with the electronic part at a volume resembling that of performance. But even more disturbingly, there is often a disconnection between the written score and the electronic part, often with chronometer markings being the only indication of a link between the two. All of the pianists involved in the creation of *Maly velky Svet* were highly experienced chamber music players, and I wanted to tap into their skill set in creating a unified sound between their actions at the piano and the electronic part. Clearly I needed an interface that would allow them to rehearse with the electronics as often as possible, even if without the ideal sound reproduction setup. At the bottom of each hyperscore, there is therefore a link to the videoscore of the work, which integrates the different parts of the piano notation with a timer and the fixed audio part, which includes a mockup of the live electronic elements. Two of the three performers were very enthusiastic about this tool, which allowed them to incorporate the sonic landscape of the work in their everyday practice. They reported that they did not miss a visualisation of the electronic part because the timings that were shown were always accompanied by the sonic event to which they referred, eschewing the need for a visual explanation. At some point we experimented with the integration of a waveform or spectrogram, but since these performers were not adept at or interested in reading these and found their presence more obtrusive than helpful, I kept the information on the screen at any moment as reduced as possible. The notion of showing the minimum information needed came from interviews I had conducted with performers of electroacoustic music in 2012 abot notation and collaboration [6]. Since then, however, I have started wondering whether there is not a potential instructive value to including notation that has become conventional to musicians working with digital media. Subsequent videoscores, discussed in the next section, which I created for my own use flip the type of information included around: instead of traditional notation and a timer, I favour the use spectrograms and waveforms.

2.3 Paperscores

At this point, a conventional score that can be printed is still necessary for most performers. While an increasing number are using tablets or digital devices, most performers still want to have a copy to mark up and use for technical practice of isolated passages. To my surprise, when we came to the final rehearsals for *Maly velky Svet*, several months after the pianists had received the various score formats, one of the three arrived with a tattered printed copy that she had been consistently using and admitted that she only rarely gave the videoscore a go. Clearly some habits – and notational media – die hard.

3. SCORES FOR ELECTROACOUSTIC PERFORMERS/IMPROVISERS/ COMPOSERS

On some level I understand the reluctance by classically-trained instrumentalists to learn to interpret a new form of notation. I had been looking at spectrograms and waveforms for several years in my electroacoustic compositional practice before I thought to use them as a form of notation. Perhaps this springs from the fact that electroacoustic notation and the software available, ranging from audio editors to musical analysis tools, all work on a descriptive model: they aim to visualize the sound rather than prescribe what to play. There seem to be few options for doing both at the same time, which means that users such as myself, who want to have access to the visualization of fixed media elements while at the same time show prescriptive (Western) instrumental notation, and who do not want to develop new software, resort to hacking or combining. Perhaps this is because there are not that many instrumental performers (and/or composers) versed or interested in learning and using such tools, or because such combined visualizations might become very messy or complicated for works with more than just a few performers.

In any case, it was only when I wanted to notate works I would play myself, where I did not have to worry about any one else's learning curve, that I began to experiment with including prescriptive notation within descriptive/analytical software such as the GRM Acousmographe or Pierre Couprie's EAnalyis. The creation of two long works for recorders and electronics with Monty Adkins and Hildegard Westerkamp seemed the perfect opportunity to try out some new strategies without having to worry about transmitting all the small details of performance practice. Both Adkins and Westerkamp were more than happy to leave the bulk of the decisions and most importantly the notation of the live recorder part to me. This was entirely logical since these works were largely focused on sounds and playing techniques that were highly specific to my instruments and idiomatic to my way of playing them. In both collaborations, it was also clear that these were not works intended for any other performer to play, they relied on my creating them in the moment every time anew, my playing being an integral part of the work itself. I was therefore in the best position to know how to create the right mnemonic device.

3.1 Lepidoptera

In *Lepidoptera* (2014-15), a 40 minute, five-part work I wrote with Monty Adkins, we shared the recording, editing, processing and putting together of the electronic material. Once that was established, I largely improvised the live recorder part and the playback and processing was often determined with some degree of aleatory by the computer. The indeterminacy was highly controlled by predetermined sets of playing techniques, samples, and a fixed timeline. Each movement relied on a different strategy and what I played live evolved and became increasingly specific – though not fixed – over the course of the compositional and performance process. At first, when we were composing the work, I needed a score simply for cues and the simplest instruction, since I could remember most of my decisions from one time to the next. Right after we finished this intensive compositional period, we recorded the work, thus creating a document that would then end up serving as my aide-memoire for the bulk of our performances a year later. I was very thankful for that recording, since it allowed me to reconstruct the work – and create a first real score for performance.

This score was a hybrid between an electronic performance patch in Ableton Live and five videos, one for every movement. For each movement I needed different information – sometimes just a descriptive score of the fixed media in the works where my part was entirely free, other times screen shots of the pertinent parts of the patch in action. The second movement, *Lepidoptera,* has the most "fixed" recorder part, meaning that I play specific fingerings and techniques at determined moments (to align with the automated parameters of the live processing). I had made the live processing in studio using a mockup of my live part and so I used an Acousmographe of that same mockup to make the performance score. I used the Acousmographe because I liked the control I had over the visual aspects of both the waveform and spectrogram. I read the waveform to give me information about the timing and shape of the notes I was to play and the spectrogram to show what harmonic content/richness I should aim for in the multiphonic/overblown fingerings. The pitch content was determined by the note fingerings I marked, as seen here in figure 1.

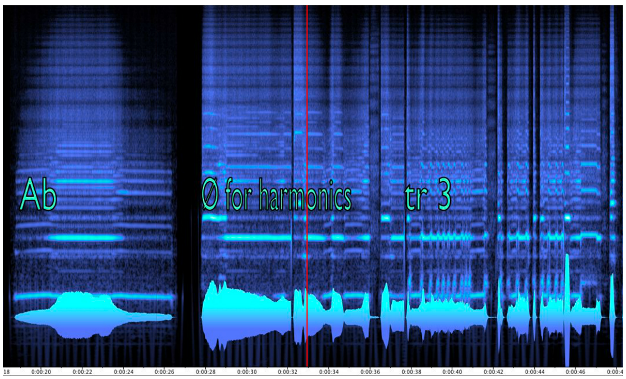


Figure 1. Screenshot from videoscore for *Lepidoptera*. @ 2015 Terri Hron.[7]

The Acousmographe is used here simultaneously as a descriptive and prescriptive tool: it shows what I have to do by using an example of what I have done in the past. Clearly I do not recreate the waveform exactly, but as I perform and rehearse it, I have learned to become increasingly exact. This has then led me to question whether I actually intended such exactitude in the first place, or whether the tool has led me to become so.

3.2 Beads of Time Sounding

The next score for which I used the Acousmographe was *Beads of Time Sounding* (2016), a piece which I wrote with Hildegard Westerkamp that can range from ten to sixty minutes. This collaboration with Westerkamp was based on a series of recordings that she made of me in 2010, improvising in locations significant to her childhood in Osnabrück, Germany. I defined the material I played over the course of the three recording sessions in terms of instrument choice and technique, placing myself as a set of soundmarks in these different locations. When it came time to create the work, I let myself be guided by Westerkamp's deep experience of soundscape composition, and her preference for fixed pieces.
In the beginning, Westerkamp assumed I would simply improvise over and within the soundscape "beads" we would create, since she was at a loss for how to set down or notate anything more specific. She met my suggestion that I use the Acousmographe to create a more detailed score – even if I were to be largely improvising my live part – with enormous enthusiasm, since it was a tool that allowed us to discuss and talk about the sound in a very specific, electroacoustic way. Unlike the score for *Lepidoptera* shown above, the score for *Beads* does visualize the part I should play, but rather the output of the fixed audio part. In *Lepidoptera*, the electronics are different every time, since they are put together in real time from a discrete sets of samples, and the fixed part is what I play. In *Beads,* I am responding to the fixed electronics in the space with the instruments and techniques specified in the score, as shown below.

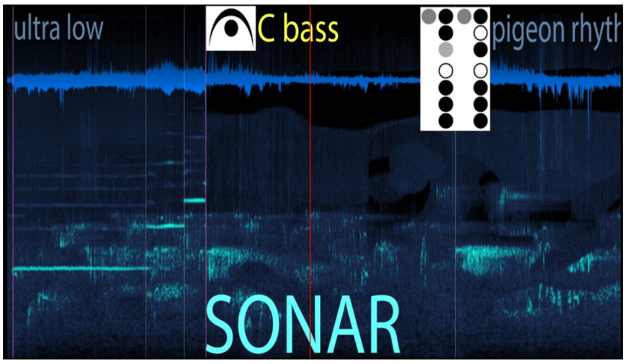


Figure 2. Screenshot from videoscore for *Beads of Time Sounding*. @ 2016 Terri Hron.[7]

One of the features of this piece is the echoing of my playing as recorded in the original field recording by what I play live. As such, the harmonies/spectra and textures that I am creating with the (overblowing of) fingerings shown are implicit in the sound I am playing with. What I need to see are the important and less important cues to synchronize with and the general form and progress of the piece.
The performances of *Beads of Time Sounding* have convinced me of the viability and power of reading the fixed part in this way (and the videoscore is perfect for practice with the integrated audio as well), since it offers me very precise synchronicity.

3.3 CARDIAC

The ease and simplicity of using the Acousmographe to sync with a fixed audio part motivated me to use it in my latest work, *CARDIAC* (2016)*,* for other performers experienced in playing with electronics: the violin/piano duo Wapiti. After discussing various options that ranged from a videoscore integrating a *Beads*-like Acousmogrpahy with traditional staff notation (like the *Maly velky Svet* videoscores) to a simple paperscore with cue timings, we settled on a hybrid system with a videoscore Acousmography that showed a visualisation of the fixed audio with cues referred to by the paperscore. The performers chose this option since they can read the video on their small phone or tablet devices while still having a full-size staff notation. On my side, it gives me a chance to provide both a reliable cuing/sync mechanism and a description of the electronics. The performers have already been very enthusiastic about how much information this dual system provides. In a sense, we are back at a kind of multiple – or in this case hybrid – format for the different needs, practical and technological, of performers working with media.

4. CONCLUSIONS

Electroacoustic music for instrumental performers has catalyzed the use of notation beyond the staff or the page to deal with the issues that digital media offers and imposes. These include a relationship to fixed (digital) timelines and a greater focus on the extended exploration of spectra and texture. Conversely, the performer and her need for effective rehearsal tools force electroacoustic practice to develop adequate means as well. The non-standardized nature of the relationship between performer and media has eluded a single notation tool, favoring instead a flexibility and fluency with many. It is my hope that the experiences and ideas described here can be a springboard for discussion of performer-oriented electroacoustic notation practice.

5. REFERENCES

[1] L. Goehr, *The Imaginary Museum of Musical Works: An Essay in the Philosophy of Music.* Oxford, 1992, especially 243-286.

[2] T. Hron, *Composing idiomatically for specific performers: collaboration in the creation of electroacoustic music.* DMA dissertation, University of Montreal, 2015.

[3] U. Eco, “The Poetics of the Open Work,” in C. Cox & D. Warner (Eds.), *Audio culture: readings in modern music*. Continuum, 2004, p. 172.

[4] W. Kentridge, *Six drawing lessons*. Harvard University Press, 2014, p. 87.

[5] T. Hron, *Maly velky Svet: FATE,* http://www.birdonawire.ca/mvs-fate-hyperscore/ and *Maly velky Svet: KNOCKING,* http://www.birdonawire.ca/mvs-knocking-hyperscore/

[6] ibid. Personal interview with Dana Jessen, April 24, 2012. Discussed in “Exploring the nexus of collaboration, notation and meaning in mixed electroacoustic music,” in in *Proc. Electroacoustic Music Studies Conference*, Stockholm, 2012.

[7] INA-GRM. *Acousmographe*. Computer software, 2014. http://www.inagrm.com/accueil/outils/acousmographe.

PERFORMER ACTION MODELING IN REAL-TIME NOTATION

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

ABSTRACT

This paper discusses the application of action-based music notation, and in particular performer action modeling, to my real-time notation (RTN) work, *Terraformation* (2016–17), which uses a combination of common practice notation (CPN), fingerboard tablature, and color gradients.

1. INTRODUCTION

Physical gestures are perhaps the oldest form of human communication, predating vocal language. Recent anthropological research points to the universal phenomenon of manual sign languages and their ease of adoption by infants to suggest that such gestures were the primary communication mode of early bipedal hominins [1]. Similarly, the notation of manual action precedes any notation resembling common practice notation (CPN). Clay tablets dating to the Old Babylonia period (ca. 2000–1700 B.C.E.) depict scales on a four-stringed lyre using cuneiform tablature notation, arguably making action-based music notation the oldest form of music notation [2].

While tablatures for specific instruments (lute, guitar, organ, etc.) [3] have existed for centuries, a generalized approach to action-based music notation has only been attempted in the twentieth century. For centuries before, CPN focused on notation suitable for describing the resultant sound. Action notation is typically subsumed under the more general category of graphic music notation or text-based music notation, both of which act as extensions or replacements of CPN. These additions and expansions developed concurrently with similar trends in the visual art world. This paper will describe several ways composers have notated performer action rather than resultant sound.

Action-based music notation is a viable solution for a major problem in real-time notation (RTN), namely the need for efficient notation in order to facilitate quick and accurate sight-reading. "Pure action-based scores in fact utilize images that suggest clear instructions at first sight and need no further explanation. Such scores could literally be sight-read" [4, p. 67]. My RTN work, *Terraformation* (2016–17) for viola and computer, uses a combination of action-based notation and CPN [5]. The action-based elements are generated from a model of the physical actions required to produce sounds on the viola. The notation is designed to evoke complex and expressive musical outcomes while being as visually efficiently as possible. In this way, I propose that the application of action-based notation to RTN is both a fruitful extension of the action-based experiments in notation and a solution to one of the key problems of real-time composition.

2. NOTATING ACTION

Music notation mediates the relationship between composition and performance. Expansions of notational language correspondingly expand and modulate those relationships. The following discussion explores different expansions of CPN through the addition of abstract graphics or textual direction and their effect on compositional process and performance practice.

2.1 Resultant Sound Notation

Many notations have been developed through the twentieth and early twenty-first centuries, but not all of them refer to action. Like CPN, some notations invent new ways of notating resultant sound. John Cage's score for *Aria* (1958), for instance, uses line contours plotted on a Cartesian pitch/duration axis colored in such a way as to represent different styles of vocalization [6]. The notation uses symbols distilled from CPN to address traditional parameters of music rather than performer action.

Karlheinz Stockhausen's *Plus Minus* (1963) is another example of new notation that only addresses the resultant sound [7]. The score for *Plus Minus* asks the performer to construct the details of the piece by reconciling a complex set of instructions with several pages of abstract graphics. The work is a set of instructions for making an indeterminate number of compositions based on the number of performers and order in which the graphics are combined. Like *Aria*, Stockhausen's use of graphics and text is directed toward musical parameters like pitch, duration, tempo, dynamics, and articulation rather than performer action.

2.2 Performative Action in Notation

One of the earliest forms of performative action in CPN can be traced to textual stage directions in theatrical works [8]. Before that, several types of action-based notation existed for the purpose of communicating and preserving dance choreography [9]. Many experimental notation systems in the twentieth and twenty-first centuries ask the performer to engage in detailed bodily or instrumental action. The range of action techniques and notational language demonstrates the variety of reasons for such use: music as theater, sound

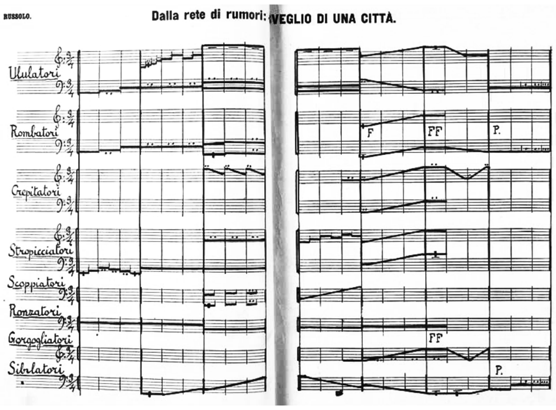



Figure 1. Line contour notation in Luigi Russolo's *Risveglio di una città* (1913–14) indicating the crank speed, pressure, and resulting dynamic of *intonarumori* instruments.

production, indeterminate parameters, notational efficiency, intentional complexity, or performer freedom to name a few.

The notation of some actions is directly correlated with playing. This is often the case when writing for a new instrument without an established tradition of performance practice. Luigi Russolo in *Risveglio di una città* (*Awakening of a City*) (1913–14), for example, notates the speed, pressure, and resulting dynamic of his crank-driven *intonarumori* instruments [10] (Fig. 1). Russolo combines familiar CPN elements like five-line staves and time signatures with graphic line contours similar in appearance to Cage's line contours for *Aria*. The difference, however, is that Cage's contours implicitly rely on the interpretation of musical parameters while Russolo's notations act as instrumental tablature.

Like Russolo, Helmut Lachenmann graphically notates action in his scores for the purposes of sound production. His *Pression* (1969, rev. 2010) [11] and *Gran Torso* (1971–72) [12] employ a mixture of CPN and tablature notation in order to explore new instrumental sounds in his pursuit of *musique concrte instrumentale* [13]. Lachenmann's introduction of the "bridge clef" and "string clef" enable the notation to directly mediate a non-standard action on the instrument. The resulting sounds of Lachenmann's actions are innately connected with the action required to produce the sound. The sound of ricochet bowing, for example, is impossible to produce using any other technique. Actions themselves are sometimes unintuitively related to the resulting sound. In his 2010 revision of *Pression*, "action dynamics," notated as dynamics in quotation marks, suggest the physical force of an action required to produce a sound with a disproportionate dynamic outcome.

The discrepancy between action and sound and the discovery of new modes of sound production is a hallmark of Aaron Cassidy's work [14]. He accomplishes this by decoupling and modulating a large set of action parameters in tablature notation. In his indeterminate string work *The Crutch of Memory* (2004) [15] and his Second String Quartet (2009–10) [16], Cassidy loosely specifies pitch information by providing the performer with a graphic contour of left hand position, variable finger width, and fingerboard location. Hand positions, fingerings, and pitches become less precise and more gestural as a consequence of this unusual approach to notating the left hand (Fig. 2).

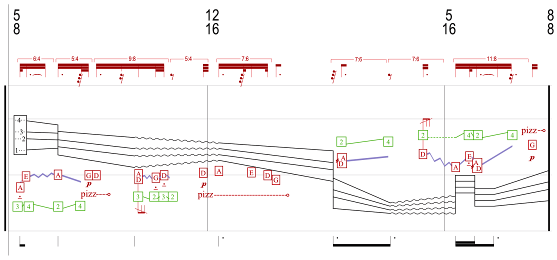


Figure 2. Decoupled string tablature in the second violin part from Aaron Cassidy's Second String Quartet (2009–10).

2.3 Cognitive Attention Balancing

One reason a composer might employ action-based notation is for the purpose of cognitive attention balancing. This constitutes an admission by the composer that each parameter addressed in the notation requires a portion of the performer's finite cognitive function. The more parameters specified in the notation, the higher demand required of the performer's brain.

Due to the limitations of CPN, action-based notation is a potential solution to simplifying performance instructions. One might imagine how cumbersome Juraj Kojs's directions in *Revelations* (2005) to scrape, bounce, and roll a variety of circular toys across resonant plates would be if notated in CPN [17]. The opposite position, that action-based notation requires more attention from a performer, is also plausible. Take, for example, Lachenmann's use of invented clefs. Tablature notation such as the bridge clef or string clef has the potential to ignore or subvert a performer's highly developed skills of reading CPN and playing their instrument. In some regards, very little prior knowledge of notation and performance technique is required or even relevant. Contemporary experiments in tablature intend to question the validity of CPN and traditional performance practice itself; this posits a potentially oppositional relationship between composer and trained performer, which is itself a determinant of the musical result.

It comes as no surprise, then, that through notation some composers purposely create a work of staggering difficulty, overwhelming the performer with a multitude of (sometimes contradictory) tasks. This is often the case in the works of Brian Ferneyhough, Richard Barrett, and others composing so-called complex scores, and is almost inevitable in the decoupled notations of Cassidy and others. The opposite situation of requiring very little specific parameter control from the musician leaves room for performers to interpret, improvise, and interact with other performers. There is evidence of this in the text-based works of John Cage — such as *Empty Words* (1974) [18] —, the group improvisation pieces of Christian Wolff — such as *For 1, 2, or 3 People* (1964) [19] —, and jazz lead sheets. Here the

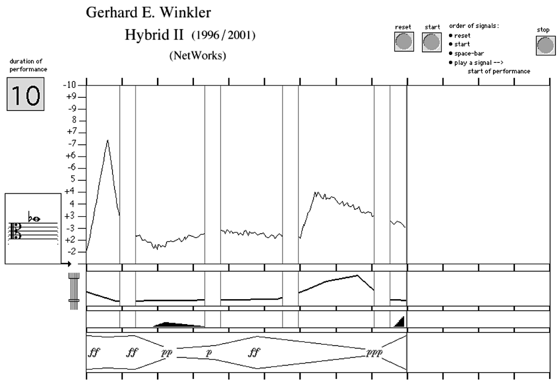


Figure 3. Line contour notation in Gerhard E. Winkler's *Hybrid II (NetWorks)* (1996, rev. 2001) indicating glissandi, bow contact position, and dynamic profile.

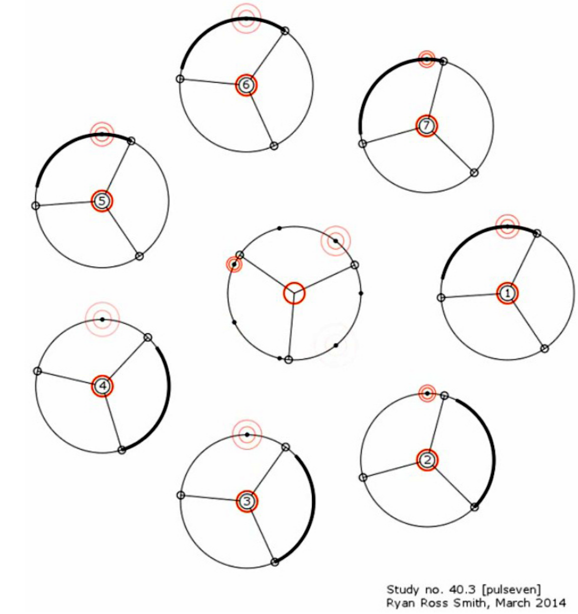


Figure 4. Radial notation indicating attack and sustain points in Ryan Ross Smith's *Study no. 40.3* (2014).

composer relies on the performer's creative abilities to collaboratively complete the music. A wide breadth of creative work lies between the extremes of notational vacuum and parameter overload, with composers often attempting to balance one difficult parameter by making the other remaining parameters correspondingly easier. This is my approach to action-based notation in my work *Terraformation*.

3. PERFORMATIVE ACTION IN REAL-TIME NOTATION

3.1 Purposes of Action-Based Notation in Real-Time Notation

Many of the earliest works using RTN are action-based. Gerhard E. Winkler's *Hybrid II (NetWorks)* (1996/2001), for example, uses several real-time line contours to direct the solo violist's glissandi, bow contact position, and dynamic profile (Fig. 3) [20]. Likewise, Karlheinz Essl's *Champ d'Action* (1998), uses descriptive on-screen text to direct a group improvisation [21]. The choice to use text and moving line segments was no doubt partially due to computer limitations. However, these early works reveal an attempt to streamline the notational elements in order create compelling music that is efficient to sight-read. As Winkler states, "In general a mixtures of *symbolic* (e.g. a "main-pitch") and *graphic elements* (e.g. Glissando-lines) has turned out to be the clearest way of Realtime-notation. It depends on the idea of the piece and the aesthetics of the composer, which elements these will be.... Which *aspects of playing* have to be notated up to which extend of precision (The range goes from *full realtime-notation*, — using all the "in-time" — possibilities of the computer-screen —, to *partly fixed and prenotated elements*, — e.g. rhythmic patterns, which can be prepared in advance —, up to *fully notated score-fragments*) [22, p. 3]." These first RTN works demonstrated efficient notation methods and prefigured a fascination with directing performer action in real-time.

Composers currently writing RTN pieces continue to use the techniques established by Winkler, Essl, and others. The radial scores of David Kim-Boyle [23] and Ryan Ross Smith [24], for example, which display a clock hand-like play head sweeping over attack points situated on a clock face, The intersection of two graphic elements is an immediately clear paradigm for complex rhythmic actions (Fig. 4). In their simplest form, these radial scores tell the musician when to perform an action. When duplicated to direct large ensembles, the radial score efficiently notates dense polyrhythmic textures.

When musical parameters are decoupled through an efficient graphical language, the performer is freed to focus their attention on the most musically challenging elements on a momentary basis. As described above, decoupling performative actions has the potential for revealing new modes of sound production. One drawback is that it also has the potential for increasing strain on the performer. Finding the equilibrium between these two objectives in RTN is a delicate task.

3.2 Performer Action Modeling in *Terraformation*

Terraformation for viola and computer uses action-based notation for the following purposes: for efficiency in sight-reading, to enable an interactive formal structure, and to reveal new modes of sound production. The performative actions required in the piece are based on a study of physical and psychological mechanisms at work in the musician's manual contact with the instrument. The resulting notation is carefully designed to ease the cognitive translation from graphic representation to bodily action.

The notation used in *Terraformation* resulted from an active collaboration with violist Michael Capone. His experiences and reactions in reading early versions of the work helped determine the present state of the piece. In particular, Capone helped me rank the difficulty of left hand positions and balance the weighting applied to the algorithm when moving the left hand from one position to another. He also narrated his sight-reading thought process as he correlated the different forms of notation used in *Terraformation*, re-

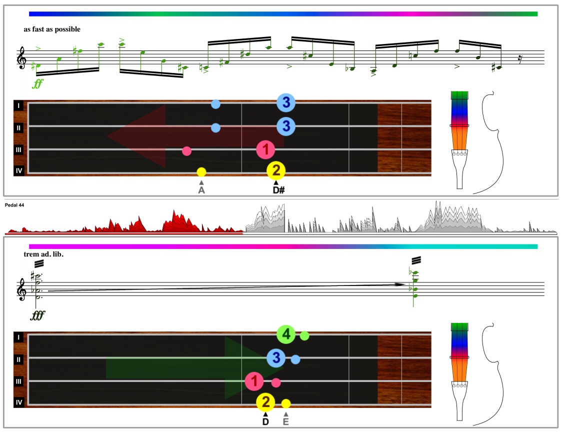


Figure 5. The aggregate notation and performance interface for *Terraformation* (2016–17).

lating when certain notations were beneficial and when they were extraneous. His guidance regarding an open-ended parameterization of the physical actions required to play the viola helped determine the parameters I chose to address in the work.

3.2.1 *Overview of Notation Used in* Terraformation

There are three distinct forms of notation in *Terraformation* (Fig. 5). One type of notation is a five-line staff with standard clefs capable of showing common music notation symbols. Elements of this staff can be hidden so that one of three different modes can be displayed at any given time: specific pitches and rhythms using standard symbols, specific pitches with proportionally spaced rhythms, or approximate pitches (displayed as stems without note heads) with specific rhythms.

The second type of notation is a tablature depiction of the viola's fingerboard. Instead of fret-like gradations of position, just the one-, two-, and three-octave positions and the approximate end of the fingerboard are marked. Each of the musician's fingers is notated on the fingerboard as a color-coded encircled number. An open or unplayed string is shown as a grayed out zero at the far left-hand side of the diagram. In addition, the lowest string with a finger down is marked with the letter name of the specific pitch for quick reference.

The third type of notation is two sets of color gradients. The first stretches across the horizontal width of the five-line staff and is used to indicate bow contact position. The specific position at any given moment corresponds to the color sharing the same vertical alignment as the current rhythm on the five-line staff. The color blue indicates molto *sul tasto*, green is *normale*, red is *molto sul ponticello*, and yellow is behind the bridge. Any gradient between those colors represents a bow contact point between the endpoints of that continuum. The second color gradient is applied to each of the note heads on the five-line staff. Ranging from black to light green, these indicate a continuum between normal left hand finger pressure to light finger pressure (as light as possibly makes a difference, slightly lighter than harmonic finger pressure).

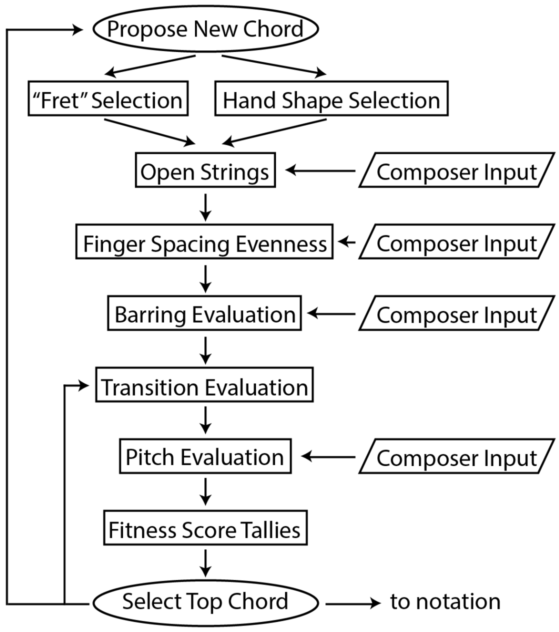


Figure 6. The algorithm for generating new chords in *Terraformation*.

These three types of notation comprise an aggregate notational system, although two of the three types are subject to display at any given moment. While the five-line staff system remains on-screen throughout the piece, the fingerboard and color gradients can be independently hidden when not required. Additionally, two aggregate systems of notation occupy the performer's screen-based score. The top aggregate system shows the notation for the current musical activity and the bottom system shows the subsequent material. Between the two aggregate systems is a graphic indicating the performers current location in the form.

3.2.2 *Fingering Positions on the Fingerboard*

The algorithm driving musical material in *Terraformation* is built on a series of constraints that model the physical action required to produce a quadruple stop on the viola, referred to hereafter simply as a "chord." The general sequence of chord creation is illustrated in Fig. 6.

This sequence of operations iterates a number of times to generate a pool of potential chord candidates. At the end of the process, the algorithm proposes the best possible choice to follow the current chord based on inputs governing the model. The action-based logic behind each of these subroutines is explained below.

3.2.3 *"Fret" Selection and Maximum Finger Stretch*

The term "fret" is used here as a method of conveniently locating the finger on the fingerboard and also as a way to avoid more conventional position-based string pedagogical practice. The model first randomly selects a fret and assigns it to the lowest-fretted finger (see Fig. 7). For all practical purposes, the lowest-fretted finger in a quadruple stop is always the first finger. Similarly, the highest-fretted finger is always the fourth finger. The exact fingers are not specified in the algorithm so as to allow for non-quadruple stopped

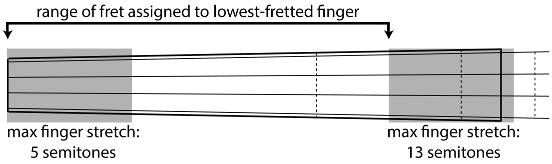


Figure 7. The range of possible “fret” positions and maximum finger stretch in those positions.

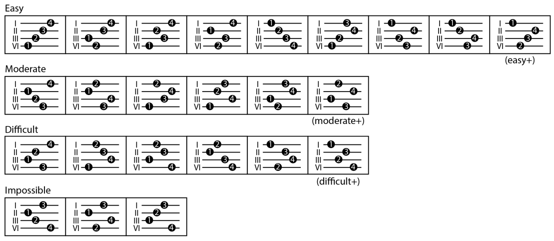


Figure 8. All twenty-four unbarred left hand positions ranked by difficulty.

possibilities where an open string or rest on the first or fourth fingers changes which finger is playing the lowest fret.

The possible range for the lowest-fretted finger is bounded on one end by the open strings at fret-0 and at the other end by fret-18. Based on the selected fret, the algorithm randomly chooses the distance between the lowest- and highest-fretted fingers. At the lowest end of the fingerboard, the maximum stretch between the lowest- and highest-fretted finger is five semitones. This stretch increases to thirteen semitones at fret-18, extending approximately to the end of the fingerboard [25].

3.2.4 Hand Shape Selection

A parallel process chooses a hand shape from a predetermined set of twenty-one options ranked by difficulty. A total of twenty-four (4! = 24) hand positions are theoretically available, but three are physically impossible (see Fig. 8).

The hand positions are selected based on a weighted transition table that prefers easier hand positions. Once chosen, the hand position is coupled with the fret selection and finger stretch parameters described above to generate a specific finger and fret combination for the lowest- and highest-fingered frets. The two central fingers’ exact positions remain to be determined. In addition, a corresponding penalty is applied to the chord’s overall fitness score based on the difficulty of the hand position. This score is tallied and ranked at the very end of the process.

3.2.5 Open Strings

Before fixing the exact fret positions of all of the fingers, the algorithm randomly allows for the selection of open strings. Open strings are applied to both finger and string, ignoring the fret parameter in the subsequent routines related to finger spacing. An input value governs the percentage chance of open strings at each chord request.

3.2.6 Evenness of Finger Spacing

The remaining two central fingers’ positions are determined in connection with an input value that corresponds to finger spacing evenness. At low-fretted positions on the fingerboard, little variation is possible for the central fingers due to the limitations of the maximum finger stretch. At higher-fretted positions, a higher concentration of pitch possibilities in condensed physical space yields more options. Two factors govern the evenness of the finger spacing. The first is a decreasing amount of flexibility between adjacent pairs of fingers from the first-second pair to the third-fourth pair. In other words, the variation in finger spacing is most flexible between the first and second fingers and least flexible between the third and fourth fingers. The second factor is that asking the player to stretch the space between one pair of fingers inhibits stretching in other fingers.

On the whole, even spacing of the two central fingers between the outer fingers is the most comfortable and therefore the more playable solution. Increasing the uneven input value randomly deviates away from even spacing using the two-factor model of finger spacing just described.

3.2.7 Barring Evaluation

At this point, the algorithm has generated a complete chord with specific finger and fret locations. Many chords are still highly impractical from a physical perspective and undesirable from a musical perspective. Several evaluation processes examine the fitness of the chord and assign it a score that when tallied rates its viability.

The first evaluation looks for chords with two fingers on the same fret, commonly referred to as barring. An input value controls whether or not two adjacent strings can be barred. Chords with three or more fingers on the same fret or with two non-adjacent strings on the same fret are immediately rejected.

3.2.8 Chord Transition Evaluation

The second stage of evaluation examines the transition between the current chord and the proposed subsequent chord. The algorithm tracks the movement of each finger from the current chord to the proposed chord and generates a score that considers the following: whether or not a finger changes strings, the direction of the move up or down the fingerboard, and the dexterity of each finger. Moving a finger from one string to another incurs a significant scoring penalty, with changes going from a higher-numbered string to a lower-numbered string being more severe than a lower- to higher-numbered string (to be clear, a lower-numbered string, ie. String I, produces higher pitches than a higher-numbered string, ie. String IV). The reason for this is that it is more difficult to contract a finger to a new position than to extend a finger. Next, the average fret positions of the current and proposed chord are compared. More distant average fret movement acquires a higher scoring penalty. Finally, each of the scoring mechanisms accounts for differences in finger dexterity by using a finger-specific weighting, with movements in the fourth finger generating higher penalties. This finger-specific weighting reflects an overall ease of movement in the first finger with each subsequent finger diminishing in dexterity.

3.2.9 Pitch Evaluation

The third stage in the evaluation process scores each chord according to a specified pitch-class, pitch-class set, or combination of both. Chords that contain one or more matches are given a higher ranking as more desirable. Each evaluation routine — pitch-class, pitch-class set, or both — can be activated or deactivated. In any given iteration of the algorithm, the pitch-class evaluator finds the most matches and the both evaluator finds the least. By requesting that chords fulfill both pc and pcs requirements, the algorithm will sacrifice ease of chord transition and playability for more desirable pitch content.

3.2.10 Fitness Score Tallies

Following generation and evaluation, a list of proposed chords are finally collected and their corresponding fitness scores tallied. The list is sorted first by chords that fit the requested pitch requirements. Within that list, chords are arrangement by the difficulty of the chord's physical production. The chord with the top score (ie. the least amount of penalties) is displayed for the musician to perform and is fed back into the chord algorithm for comparison with subsequent chord candidates. In addition, the fitness score follows the chosen chord through the creation of the remaining musical parameters — rhythmic figures, dynamic contour, bow contact position, and left hand finger pressure, to name a few. The difficulty of these parameters is inversely related to the chord's fitness score. So, for example, as the difficulty of the chord increases, the difficulty of the rhythmic figure decreases. In this way, the fitness score mediates the amount of attention that the performer is likely to spend on any single parameter.

3.3 Efficiency in Hybrid Notation

The performative action model in *Terraformation* attempts to balance the cognitive demands on the musician by using a hybrid combination of notation types. The aggregate notation display is designed to give the performer instructions that are immediately readable while also providing a depth of detail. Comments from violist Mike Capone following a rehearsal of *Terraformation* revealed the specific sequence of information gathering that he executes each time the display is refreshed. The performer first deduces the hand position from the fingerboard diagram. While he generally replicates the hand position on the instrument he is assessing the position of the lowest-fretted finger. He then finalizes hand position by checking it against the CPN, making small adjustments where necessary. The moment he spends looking at the CPN also gives him an approximate understanding of the rhythmic character of the current staff system. As he begins to perform the material, he is constantly correlating the four-color gradient that represents bow contact position and the two-color gradient that represents left hand finger pressure with the current rhythmic figure, pitch, and dynamic. Finally, in moments of minimum cognitive strain — in rests or during repeating figures, for example — he may look below the current aggregate staff system to the upcoming system in order to read ahead. In this way, through efficiency of a hybrid notation display, the musician is able to link information gleaned from different types of notation into a cohesive, continuous performance.

4. CONCLUSION

While my work modeling the physical actions required to play the viola led to the creation of *Terraformation*, this research also yields a general tool for composers writing for violins and violas. Composers, especially those without a background in string playing, spend considerable time determining the feasibility of double, triple, and quadruple stops. To solve this problem, I am developing a general tool for assessing the difficulty of any given multiple stop for violin or viola and suggesting alternative or subsequent multiple stops based on the pitch-filter criteria described above. This utility will incorporate the performative action modeling research explained in this paper to aid composers writing multiple stops in their own music.

The fascination with action-based music notation in the twentieth and twenty-first centuries has yielded a variety of alternate ways of mediating musical performance. Some of the key benefits of this category of notation include clarity of sound production techniques and immediately recognizable instructions that reduce cognitive strain on the performer. These are important factors when asking a musician to sight-read during performance as in the case of RTN. While incorporating action-based music notation into a work using RTN is not a new endeavor, the methods and benefits of doing so are still an incredibly rich area for exploration. In *Terraformation*, an algorithm modeling the physical actions required to produce sound creates, ranks based on difficulty and pitch content, and notates musical material. Finally, by using several types of notation to instruct the performer — a combination of action-based notation and CPN — the musician is able to efficiently extract and unify the instructions into a cohesive musical gesture.

Acknowledgments

I would like to thank violist Mike Capone for his guidance concerning the physical actions required to play the instrument and his insight into the mental processes involved in fast sight-reading.

5. REFERENCES

[1] M. C. Corballis, "Did language evolve from manual gestures?" in *The Transition to Language*, A. Wray, Ed. Oxford: Oxford University Press, 2002, ch. 8, pp. 161–179.

[2] The Schoyen Collection. (2016) Ms 5105: Oldest known music notation. [Online]. Available: http://www.schoyencollection.com/music-notation/old-babylonia-cuneiform-notation/oldest-known-music-notation-ms-5105

[3] C. F. A. Williams, *The story of notation*. Detroit: Singing Tree Press, 1968.

[4] J. Kojs, "Notating action-based music," *Leonardo Music J.*, vol. 21, no. 1, pp. 65–72, Dec 1, 2011.

[5] S. Shafer. (2016) *Terraformation*. [Online]. Available: http://sethshafer.com/terraformation.html

[6] J. Cage, *Aria*. New York: C. F. Peters, 1958.

[7] K. Stockhausen, *Plus Minus*. London: Universal Edition, 1963.

[8] K. Syer, "From page to stage: Wagner as *Regisseur*," in *Richard Wagner and his world*, T. S. Grey, Ed. Princeton: Princeton University Press, 2009, pp. 3–22.

[9] A. H. Guest, *Choreo-graphics: a comparison of dance notation systems from the fifteenth century to the present*. New York: Gordon and Breach, 1989.

[10] L. Russolo, *The art of noises*. New York: Pendragon Press, 1986.

[11] H. Lachenmann, *Pression*. Wiesbaden: Breitkopf & Härtel, 1969 rev. 2010.

[12] ——, *Gran Torso*. Wiesbaden: Breitkopf & Härtel, 1971 rev. 1988.

[13] ——, "Paradiese auf zeit," in *Musik als existentielle Erfahrung*. Wiesbaden: Breitkopf & Härtel, 1993, pp. 205–212.

[14] A. Cassidy, "Determinate action/indeterminate sound: Tablature and chance in several recent works," in *Facets of the second modernity*, ser. New music and aesthetics in the 21st century. Hofheim: Wolke Verlag, 2008, vol. 6, pp. 17–32.

[15] ——. (2004) The crutch of memory. [Online]. Available: http://aaroncassidy.com/music/crutchofmemory.htm

[16] ——. (2010) Second string quartet. [Online]. Available: http://aaroncassidy.com/music/secondquartet.htm

[17] J. Kojs, "The language of action and cyberaction-based music: Theory and practice," *J. of New Music Research*, vol. 38, no. 3, pp. 285–294, Sep 1, 2009.

[18] J. Cage, *Empty words: writings '73–'78*. Middletown, Conn.: Wesleyan University Press, 1979.

[19] C. Wolff, *For 1, 2, or 3 people*. New York: C.F. Peters, 1964.

[20] G. E. Winkler, "*Hybrid II (NetWorks)*—or: At the edge of musical self-organization," in *Electronics in New Music*. Hofheim: Wolke Verlag, 2006, pp. 236–249.

[21] K. Essl. (1998) *Champ d'Action*: Realtime composition environment for computer-controlled ensemble. [Online]. Available: http://www.essl.at/works/champ.html

[22] G. E. Winkler, "The real-time score: A missing link in computer-music performance," in *Proc. of the Sound and Music Comp. Conf.*, Paris, 2004, pp. 9–14.

[23] D. Kim-Boyle. (2012) Point studies no. 4. [Online]. Available: http://www.davidkimboyle.net/point-studies-no-4-20141.html

[24] R. R. Smith. (2014) Study no. 40.3 [pulseven]. [Online]. Available: http://ryanrosssmith.com/study40_3.html

[25] P. Rolland, *The teaching of action in string playing: Developmental and remedial techniques*. Urbana, Illinois: Illinois String Research Associates, 1974.

EXPRESSION MARKS FOR PROGRAMMING INTERACTIVE MUSIC

Juan Carlos Martinez Nieto
Georgia Tech Center for Music Technology
jcm7@gatech.edu

Jason Freeman
Georgia Tech Center for Music Technology
jason.freeman@gatech.edu

ABSTRACT

The present work uses common Western music notation to represent logical and systematic behaviours of computer music processes in the context of score-oriented interactive music. The algorithmic representation is described by adding programming annotations in a controlled natural language to a musical staff as expression marks in the score. We implemented a computational environment that is able to translate these expression marks into coding instructions and execute them in real-time during a live performance of an interactive-music piece. A collection of short interactive music exercises for MIDI-controlled piano based on the proposed notation was composed and edited using music engraving software. During the compilation stage, an encoded version of the score in MusicXML format is translated into scripting code, and during live performance the computational environment executes the code in real time in sync with the human-performed parts. This paper introduces the syntax of expression marks for programming interactive music through a classic "Hello World" example in the context of interactive music and explains the technical details behind the implementation of the computational environment. The main motivation behind this work was to evaluate the viability of creating a cohesive symbolic representation of interactive music that is independent of specific software and hardware frameworks, and is strongly connected with the western music tradition.

1. INTRODUCTION

In the context of score-oriented interactive music, creating and preserving repertoire is not straightforward. One reason is that performance information regarding an interactive piece is not entirely represented in a musical score, so an important part of the information resides inside the computational framework on which the piece runs. This issue creates strong dependencies with a particular technology, a factor that has made some composers move away from the computer music genre [1].

In this paper, we propose to create a cohesive representation of an interactive music piece by keeping both performance instructions for human/acoustic musicians and performance instructions for a computational process in a single music score. Logical descriptors are added as expression marks in a musical staff (see section 4) and interpreted in real time by a programming engine during a live performance. This work presents a novel approach that fuses algorithmic thinking and traditional Western music notation. This approach is particularly well suited for modeling incremental music processes that usually are present in minimalist aesthetics [2].

This work is a first step in extending the technology independent representation of music, a common ground for pure instrumental music, into the field of interactive music. Selfridge-Field asserts that "since the representation of music is entirely independent of the use of computers, there is every reason to expect that codes designed for the representation of music in computer applications will eventually be entirely independent of both hardware configurations and software processes" [3].

Expression marks in common Western music notation have been used in musical scores since the eighteenth century to represent variations of tempo, intensity, and articulation [4]. The term is misleading as the scope goes beyond expressiveness in music performance [5]. In connection with the extensibility of expression marks, it is useful to bring the definition of the *Harvard Dictionary of Music* "Symbols and words or phrases and their abbreviations employed along with musical notation to guide the performance of a work in matters other than pitches and rhythms" [6]. The multi-purpose implicit characteristic of expressive marks, along with the fact that they are text-based signs, makes expression marks well suited for the purpose of extending a musical score with an algorithmic descriptor.

In regards to score-oriented interactive music, sometimes referred as score-driven interactive music [7], temporal relationships between human-performed musical events and automatic music processes play a fundamental role when creating interactive music [8]. In musical scores representing time-relationships among discrete-time events is simple and accurate. This fact motivates us to employ common Western music notation to represent systematic behaviors in interactive music. From this perspective, the score acts as a symbolic *source code* where the composer abstracts and clearly records structural relationships in time among the different musical entities (human performed instruments or automatic computer processes).

In summary, we propose to extend common Western music notation to describe systematic procedures by adding programming annotations as expression marks. The proposed approach allows representation of both the instru-

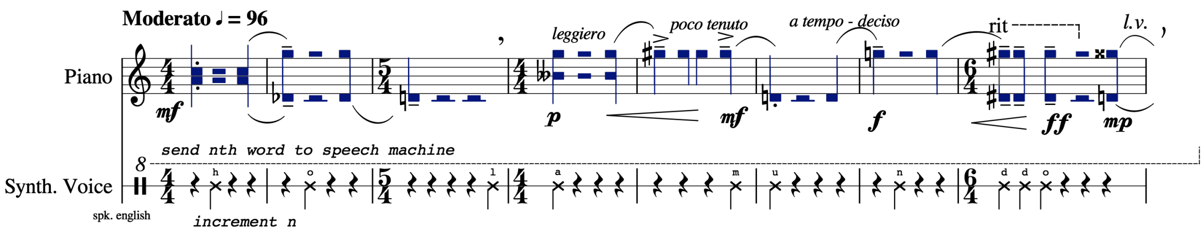


Figure 1. Score of the interactive exercise ‘Hola Mundo’.

mental music and the algorithmic process in a single and well-understood standardized document such that the music is readily readable by a human, processable by a machine, and recreatable in other systems into the distant future. In the following sections we first discuss some related work, then walk through a simple example to introduce the basic concepts behind programming-expression marks, and finally the software implementation and a case study will be detailed in section 6.

2. BACKGROUND

The present work is loosely related to score languages, which have a long tradition in the field of computer music [9]. Score language refers to a text-based list of actions arranged in absolute or symbolic time. In that sense Score language is more related to a data structure with basic programming functionality. Max Mathews developed at Bell Labs a series of score languages known as MUSIC-N in the field of audio-synthesis; the first one of those languages appeared in 1957, and the last one MUSIC-V around 1969 [10]. MUSIC-V became popular in the academic and scientific world and was extensively used in the computer music field during the second half of the twentieth century.

The Score Language pattern paradigm has been used in many music programming languages since then; mainly in the field of audio-synthesis. Common Lisp Music [11] and Csound [12] are direct descendants of MUSIC-V languages. Nowadays, Antescofo [13], one of the most popular environments for interactive music that runs embedded on Max/Msp and Pure-data environments [14], provides a text-based Score Language for describing customized actions.

Score Languages are essentially sequences of events in their core conception, so modeling high level interactions among music entities can only be done at a very basic level. Dannenberg in his survey of Music Representation Systems states that:“This approach is straightforward, but it makes it difficult to encode structural relationships between notes” [15]. Modern Score languages address this issue by embedding custom-language programming scripts. For example, RTCmix, a score language started by Lansky, includes MinC, a C-style scripting language [16] and Antescofo enables the combination of score based instructions with data structures and control-flow logic all within a single script, but with a score-following paradigm [13].

However, from the symbolic representation perspective, these approaches create two different and simultaneous models of the same music that is represented. One is the score-representation for performance and the other one is the logic-representation for the computational framework. Our work addresses the representation of music interaction in a different way, by keeping the representation of the music in a unique and cohesive symbolic source and taking advantage of the multiple semantic connotations of common Western music notation for modeling structural relationships among musical entities.

This work provides a written representation that models automatic music processes in interactive music as an extension of common Western music notation. From the machine perspective, it implies building a programming engine that is able to interpret the programming-expression marks notated in the score.

Thanks to open standards for encoding a musical score, such as MusicXML [17] or MEI [18], an encoded file version of a musical score can be understood by a machine. For the purpose of this research, the composer can model and record an interactive process directly in the score by adding programming-expression marks during the editing phase using a third-party notation software, and the encoded version of the score is interpreted by a specific purpose programming engine.

3. HELLO WORLD

This section presents the basics of expression marks for programming interactive music using the classic “Hello World” approach as a walk-through example. Figure 1 shows the score of *Hola Mundo*, a very short interactive exercise for MIDI-controlled piano and synthesized voice. The bottom staff represents the synthesized voice part that is played automatically in Supercollider [19], and the top staff shows the human-performed part that is played on the piano. The score uses square shaped note-heads only to emphasize visually the algorithmic character, and they do not have semantic meaning.

Figure 2 shows the script library that is imported by the interpreter engine at the compilation stage. The technical details about how all different levels of information are connected to be able to synchronize and execute the instructions during a live performance will be explained later in section 6. Here we will present the underlying concepts.

```
var words = [];
words[0]= 'h';
words[1]= 'o';
words[2]= 'l';
words[3]= 'a';
words[4]= 'm';
words[5]= 'u';
words[6]= 'n';
words[7]= 'd';
words[8]= 'd';
words[9]= 'o';
```

Figure 2. JavaScript code for 'Hola Mundo'

3.1 Roles in Interaction

Note that each part plays a different role in the interaction (see Figure 1). The piano part acts as control signal (input) used by the programming engine to estimate the current symbolic time-position in the score (i.e. measure and subdivisions) during a live performance. In this paper, we label this interaction type as control role, as the input controls the pace at which the instructions are executed. Furthermore, the human-performed part acts as an external synchronization clock that adjusts the internal clock of the programming engine every time that a control signal is received.

The second staff plays two simultaneous and different roles in the *Hola Mundo* exercise. First, it increments a variable in steps of one. In other words, it defines a systematic behavior, which in this Hello World exercise is constrained to increment the variable `n` every time a quarter note appears on the bottom staff. We named this interaction type as logic role. The second role of the bottom part is derived from the 'sent' expression mark text. This mark is translated in sending a message to the speech machine (synthesized voice) every time a quarter note is notated. This type of interaction represents the input parameters of an external computer music process (Supercollider in this case). We labeled this interactive role as process role.

3.2 Programming-Expression Marks

The programming-expression marks in the bottom staff of *Hola Mundo* are entered in a controlled natural language. As an example, take the expression "*send nth word to speech machine*". When the encoded version file (e.g. MusicXML) of the music score is compiled (i.e. transpiled is the proper term for describing a translation among different source codes), the compiler looks for the following syntactic text structure: *action [expression] to [output]*, then a text disambiguation operation is applied to the string literal to get each part of the text. Next, the compiler checks if the expression is a key that maps to a stored script literal and if so replaces the original expression by the mapped script version. In this Hello World exercise, the string "nth word" is mapped to the script variable *words[n]*.

This *text-to-code* mapping mechanism strongly contributes to the ease of building and preserving interactive music repertoire for the following reasons. First, the symbolic representation (staff notation) acts as a descriptor of an algorithmic behavior at a higher level, allowing the composer to abstract and record the logic of the interaction, which along with the instrumental parts should be sufficient to recreate the piece in the future without the participation of the original performers and technicians. Second, an ambiguity is introduced in the symbolic representation as the technical implementation details are not recorded in the musical score. Thus, the music can be adapted to technological and aesthetic changes in the future, avoiding that the piece being frozen in time, in a similar way as expression marks work in the context of instrumental music performance.

4. THE MEANING OF PROGRAMMING-EXPRESSION MARKS

In a general sense, a programming-expression mark is a text-based descriptor of an algorithmic function that is applied only to the specific score-part where it is defined. It is executed every time that a note appears, and it is not executed on rests. This approach enables building a parallel environment where the algorithmic functions could run at different rates as each score-part could potentially evolve independently over time.

The programming-expression marks scope can be defined to cover the whole section or to be restricted to a single note. In the first case, usually the commands are repeated until it reaches the double-bar which enables reducing the information in the score. In the second case, the command is executed just one time in that specific note which is very useful for initializations. The following paragraphs will provide more details about the syntax and semantics of programming-expression marks.

In the Hello World exercise, the expression mark "*send nth word to speech machine*" appears in the first measure. After making the text disambiguation by the parser in the compilation stage, the text is split as follows: 'send' (action), 'nth word' (identifier), 'to' (connector) and 'speech machine' (identifier). The action *send* is interpreted during live performance as sending an OSC message [20] every time a note appears. The compiler searches internally for the identifier value (nth word), if this entry exists, the compiler replaces the identifier with the stored coding expression associated with this entry. In the *Hola Mundo* score, the identifier "nth word" is mapped to the script expression "words[n]". Using a similar approach, the identifier "speech machine" is mapped to a pre-defined OSC message template. Additionally, the send action in *Hola Mundo* associates the variable "words[n]" to the content of the OSC message. Before the compilation stage, a mapping table that associates each identifier with its equivalent is added.

All actions are assumed by the compiler to be repeated every time a note appears until the music section ends (separated by double barline) except when the modifier 'once' is present. The 'once' modifier constrains the scope of the action only to the current note (in contrast to the whole section). Table 1 shows a complete list of actions for the ex-

ACTION	MEANING
send	send expression-message to port
increment	var = var + 1
compute	execute expression
assign	var = expr
jump to	goto measure
print	print(expression) in console
stop	clear all actions on the current section

Table 1. List of actions

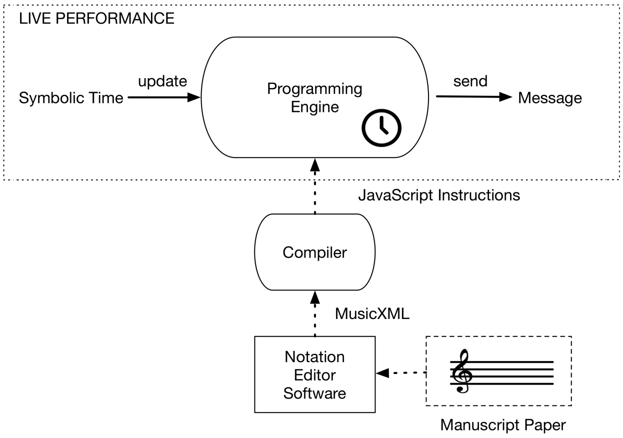


Figure 3. Implementation block diagram.

amples in this paper. The second column shows an equivalent pseudo-code of how the action is translated to the language engine during the compilation phase.

5. IMPLEMENTATION

Dynamic Programming Languages such as JavaScript are well suited for live environments as they enable interpreting and executing code in real-time. This programming approach is often referred as to Just-In-Time compilation or dynamic compilation [21]. Our implementation of the computational environment that interprets programming expression marks was written in C/C++ and has an embedded JavaScript engine to perform the Just-In-Time compilation of scripting code during live performance.

Figure 3 shows the block diagram of the actual implementation of the real-time environment based on the proposed notation for music interactive systems. The core of the implementation is the programming engine that interprets the programming expression marks in the score during a live performance. We use the music notation editor software *MuseScore* [1] for creating, editing, and exporting the score to MusicXML format.

The first step is to compile (transpile) the encoded version of the musical score. This step involves text disambiguation of the programming-expression marks in the score and translation of these commands into machine instructions. During this compilation stage, an encoded version of the score in MusicXML format is mapped to an intermediate scripting version in JavaScript that is stored in the programming engine, and it contains the symbolic-music-time locations where the instructions should be executed. Now the environment is ready for execution.

The programming engine has an internal clock that estimates the current symbolic music time, and based on that time, the corresponding scripting instructions are executed. As shown in the block diagram, an external signal with the current symbolic time feeds the programming engine to update the internal music-symbolic-time clock. Based on this update the internal music-time is estimated. This external input signal is derived from the live performance of the control-role parts(i.e. the human performed parts). In our implementation the external control signal is received via OSC. Furthermore, as shown on the block diagram, the output of the system consists of OSC messages that are sent to an external computational music framework.

6. CASE STUDY

A collection of short interactive exercises for MIDI controlled piano were composed to evaluate the viability of the proposed notation. Figure 4 shows the architecture of the implemented environment. In this setup, a human performed digital piano sends MIDI messages to a simple score following system implemented in Objective C that essentially detects chords events. The score following system estimates the current symbolic time position in the score, and sends the value to the programming engine via OSC.

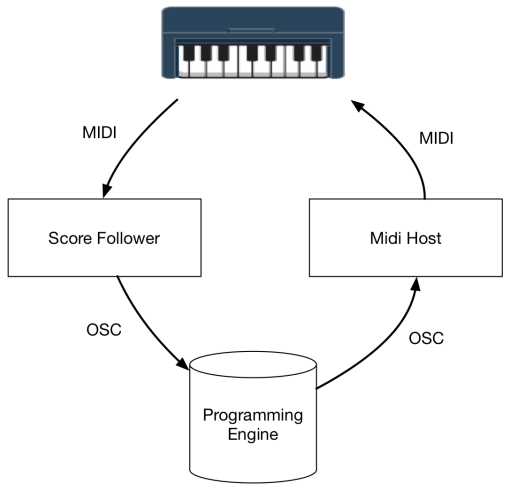


Figure 4. Implementation diagram.

As shown in Figure 4, the interactions in these piano exercises are focused on enhancing the human performance by adding an automatic counter-part played by the MIDI-Host application. The interactions are in essence minimalistic but in the variety of process music [22], meaning that one of the parameters of a music entity is gradually changed, and it is the process itself which determines the overall form of the piece [2]. Furthermore, this minimalistic approach to music composition is well-suited for evaluating a symbolic representation of logical behaviors.

[1] https://musescore.org/.

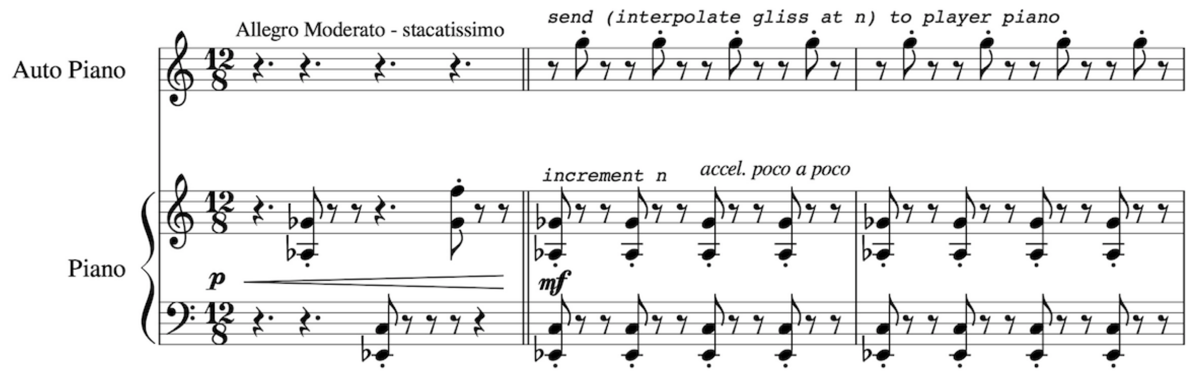



Figure 5. Score of ‘Cencerro Deslizante’.

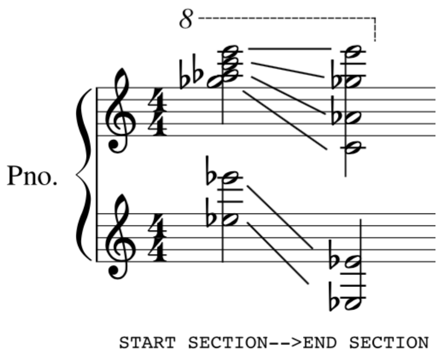


Figure 6. Process description for ‘interpolate gliss at nth’.

During a live performance, OSC string messages are sent from the programming engine to a MIDI Host application developed in Objective C. The messages are strings in JSON format, and they contain the chord notes to be played by the MIDI Host Application. Note that the MIDI Host does not add any logical layer to the interaction environment. The JSON messages are mapped to MIDI messages in the MIDI Host controller application and played back in the Digital Piano. In this context the digital piano behaves as a hyper-instrument.

Figure 5 shows the first three measures of `Cencerro Deslizante`, one of the exercises of the collection, and Figure 6 shows a segment of the performance notes of this interactive exercise. In this exercise, the automatic piano part plays off-beat chords computed from an incremental process that interpolates between two chords. Each incremental process runs over a complete section (double barline) of the piece. It is explained in the performance notes and notated in the score by the programing-expression mark *‘interpolate gliss at nth’*.

7. CONCLUSIONS

This research shows that it is possible to implement a programming engine that understands a cohesive score representation of interactive music that is independent of any computational framework by extending music notation to an algorithmic context. The present work proposes a new compositional approach that does not intend to be applicable in all cases of scored interactive music. Instead, it introduces a new compositional mechanism for interactive music which is strongly connected with traditional practices of writing music through notation and takes advantage of the multi-functional semantic scope of expression marks. We will focus our future research on developing a cohesive representation of interactive music by defining a formal syntax of programming-expression marks and creating a broad set of pieces to explore and enrich the different dimensions of the introduced compositional practice. This approach is well summarized by the following statement of Roger Dannenberg: “Music evolves with every new composition. There can be no ’true’ representation just as there can be no closed definition of music” [15].

8. REFERENCES

[1] J. C. Risset, “Composing in real-time?” *Contemporary Music Review*, vol. 18, no. 3, pp. 31–39, 1999.

[2] S. Reich, “Music as a gradual process,” in *Writings on Music, 1965-2000*. Oxford University Press, 2002.

[3] E. Selfridge-Field, *Beyond MIDI: the handbook of musical codes*. MIT press, 1997.

[4] R. Rastall, *The Notation of Western Music: An Introduction*. JM Dent and Sons, 1983.

[5] L. Treitler, *Reflections on musical meaning and its representations*. Indiana University Press, 2011.

[6] D. M. Randel, *The Harvard dictionary of music*. Harvard University Press, 2003, vol. 16.

[7] J. Drummond, “Understanding interactive systems,” *Organised Sound*, vol. 14, no. 2, pp. 124–133, 2009.

[8] M. Stroppa, “Live electronics or live music? towards a critique of interactio,” *Organised Sound*, vol. 18, no. 3, pp. 41–77, 1999.

[9] G. Wang, “A history of programming and music,” in *The Cambridge Companion to Electronic Music*. Cambridge University Press, 2007.

[10] M. V. Mathews, J. E. Miller, F. R. Moore, J. R. Pierce, and J. Risset, *The technology of computer music*. MIT press, 1969, vol. 9.

[11] H. Taube, “Common music: A music composition language in common lisp and clos,” *Computer Music Journal*, vol. 15, no. 2, pp. 21–32, 1991.

[12] B. Vercoe and D. Ellis, “Real-time csound: Software synthesis with sensing and control,” in *Proc. of the Int. Conf. on Computer Music ICMC 1990*, Scotland, 1990, pp. 209–211.

[13] A. Cont, “Antescofo: Anticipatory synchronization and control of interactive parameters in computer music,” in *Proc. of the Int. Conf. on Computer Music ICMC 2008*, Belfast, 2008, pp. 33–40.

[14] M. Puckette, “Pure data: another integrated computer music environment,” in *Proc. of the Int. Conf. on Intercollege Computer Music Concerts*, Tachikawa, 1996, pp. 37–41.

[15] R. Dannenberg, “Music representation issues, techniques, and systems,” *Computer Music Journal*, vol. 17, no. 3, pp. 20–30, 1993.

[16] D. Topper, “Rtcmix for linux (part 1),” *Linux Journal*, vol. 78, no. 1, p. 5, 2000.

[17] M. Good and G. Actor, “Using musicxml for file interchange,” in *Proc. of the Int. Conf. on Web Delivering of Music, IEEE WEDELMUSIC-2003*, Leeds, 2003, p. 153.

[18] R. Perry, “The music encoding initiative (mei),” in *Proc. of the Int. Conf. on on Musical Applications Using XML - MAX 2002*, Milan, 2002, pp. 55–59.

[19] J. McCartney, “Rethinking the computer music language: Supercollider,” *Computer Music Journal*, vol. 26, no. 4, pp. 61–68, 2002.

[20] M. Wright and A. Freed, “Open sound control: A new protocol for communicating with sound synthesizers,” in *Proc. of the Int. Conf. on Computer Music -ICMC 1997*, Thessaloniki, 1997, p. 10.

[21] J. Aycock, “A brief history of just-in-time,” *ACM Computing Surveys (CSUR)*, vol. 35, no. 2, pp. 97–113, 2003.

[22] I. Quinn, “Minimal challenges: Process music and the uses of formalist analysis,” *Contemporary Music Review*, vol. 25, no. 3, pp. 283–294, 2006.

TIMED SEQUENCES: A FRAMEWORK FOR COMPUTER-AIDED COMPOSITION WITH TEMPORAL STRUCTURES

Jérémie Garcia
Universit de Toulouse – ENAC
Toulouse, France
jeremie.garcia@enac.fr

Dimitri Bouche, Jean Bresson
UMR STMS
IRCAM-CNRS-UPMC Sorbonne Universités
Paris, France
{bouche,bresson}@ircam.fr

ABSTRACT

The software framework we present implements a simple and generic representation of the temporal dimension of musical structures used in computer-aided composition software. These structures are modeled as ordered sets of abstract "timed items" whose actual dates can be set and determined following different strategies. The timed items can be linked to an underlying action scheduling and rendering system, and can also be used as temporal handles to perform time stretching and hierarchical synchronization operations. A graphical user interface associated with this model can be embedded as a component within musical editors. We give several examples of musical objects implemented in this framework, as well as examples of time-domain operations and user interactions.

1. INTRODUCTION

Contemporary music composers commonly use computer systems to generate musical structures (scores, sounds, control data for signal and media processing). Most of these structures embed a fundamental notion of timing, which is expressed differently depending on their nature, on the tools used to create or manipulate them, or on the approach of the composer working and producing them. Working with time in consistent and efficient ways is therefore an important and challenging issue in computer music practice and research [1, 2].

Let us take the example of a compositional process involving the control of sound spatialization and the motion of sound sources defined with a set of 2D- or 3D-trajectories — sequences of pairs {*time, 3D-position*} for the different sound sources. Composers can face here several non-trivial time-related tasks such as the inner timing of the trajectories, their synchronization with the content of the sound sources or with other trajectories, etc. They may also want to integrate these trajectories within a higher-level time-structure along with other musical data — typically, in a score or in a Digital Audio Workstation. Besides, all these operations are likely to be repeated and spread at different levels of the compositional process, and applied to other types of musical objects with similar temporal characteristics.

Various approaches have been explored to unify and synchronize temporal dimensions of signal and symbolic structures in computer music environments, either from a graphical point of view [3] or from a more formal, logical perspective using temporal constraints [4], however without focusing explicitly on user-interaction.

From a composer's perspective, Stroppa and Duthen [5] introduced the concept of *pivots*, as "virtual anchors that are used to describe temporal objects [which] act as a skeleton of the temporal structure of the object and will be used to organise various elements together". Bel [6] followed a similar approach to describe musical objects and satisfy temporal constraints, but used a simpler model using a unique *pivot* for each object, thus not capturing any possible complexity in the internal time structures.

The present project takes place in the OpenMusic computer-aided composition environment [7]. It originates from previous work on the control of sound spatialization processes [8], and aims at facilitating the creation of musical objects by providing a unified framework to support both computational and graphical interaction for timing operations.

We propose a generic representation and software framework designed as a "temporal backbone" for musical structures, representing them as simple sequences of timed components, which we call *timed-items*. Timed-items can have different roles to make for explicit structuring, time transformations, or synchronization operations. They help handling time both at the micro level, to specify the inner structure of the musical objects, and at the macro level, to organize them in compound structures.

We describe the architecture and the main characteristics of the model in Section 2. We then present the corresponding graphical user interface and interactions in Section 3. In Section 4, we give several examples of musical objects and interfaces implemented in OpenMusic. In Section 5, we describe internal and external synchronization operations, and how this representation system supports the definition and manipulation of global time structures in a compositional process.

2. MODEL AND ARCHITECTURE

2.1 Timed Sequences

We consider timed musical objects as subclasses of an abstract superclass that we call a TIMED-SEQUENCE. This class holds an ordered set of "timed items" which represent basic components determining the temporal organisation inside the musical object. Public accessors allow the user (or programmer) to deal seamlessly with this list of items. Subclasses of TIMED-SEQUENCE can either use these accessors to maintain it during the life of an instance, or redefine them in order to redirect reading/writing and timing operations to existing attributes of the structure. A couple of additional methods can also be overridden to redefine or complement the addition or deletion procedures of timed items in a TIMED-SEQUENCE.

The main application programming interface (API) contains the following functions:

- **get-timed-item-list:**
  ⇒ *returns the list of timed items of a* TIME-SEQUENCE
- **set-timed-item-list:**
  ⇒ *sets a new list of timed items in a* TIME-SEQUENCE
- **make-timed-item-at-time:**
  ⇒ *returns an item created at a given date in a* TIME-SEQUENCE
- **remove-timed-item:**
  ⇒ *removes an given item from a* TIME-SEQUENCE

2.2 Timed Items

The class TIMED-ITEM is also provided as a default superclass for the elements in a TIMED-SEQUENCE. Each item has a time value (or *date*), which can be explicit (specified by the user) or implicit. It is not mandatory indeed, that all the elements in a TIMED-SEQUENCE have an explicit date: if we consider a trajectory describing a motion in space, for instance, a global duration might be specified for the motion, without a specific date assigned to every point in that trajectory: in this case, points' timing can remain implicit.

If its date is implicit, however, a TIMED-ITEM must be included in a sequence where an explicitly-dated item is present before and after it. The computation of a date is then possible provided a measure of distance exists between two items of a given type, which allows for time interpolation. [1] Implicit dates are therefore computed "on-deman" (as in a lazy-evaluation approach) and are implicitly updated by any change in the TIMED-SEQUENCE.

Finally, a special tag (*master*) can be assigned to the TIMED-ITEMs, meaning that they can be used as anchors for global time transformation or synchronization operations. Only explicitly-timed items can be tagged as *master*, and the first and last items of a sequence are always considered as *master*: they have a date assigned by default, and are privileged anchors to stretch, compress or synchronize the musical structures.

[1] The same measure of distance, in the case of spatial trajectories for instance, can be used to determine if two successive items are at the same position (distance = 0). This implies an idea of stability or steadiness of the structure over the duration defined by the two items.

2.3 Time-modification of Items

Interactions with the TIMED-SEQUENCE mostly consist in setting the date of the TIMED-ITEMs by different means: graphical or algorithmically. Depending on the type of the item, this operation will yield different effects as illustrated in Figure 1:

- Setting the date or moving an "untimed" item makes it explicitly *timed*.
- Changing the date of an explicitly timed item affects and updates the implicit date of its direct and indirect untimed neighbours.
- Changing the date of a *master* item changes the date of its direct or indirect timed neighbours and thus impacts the implicit date of their direct and indirect untimed neighbours as well.

The set of possible types of item: $\{untimed, timed, master\}$ is therefore strictly ordered, which introduces a hierarchy in the temporal structure that influences the way explicit or implicit dates are computed: the date of an item is systematically updated according to the dates of its closest left and right neighbours of a superior type.

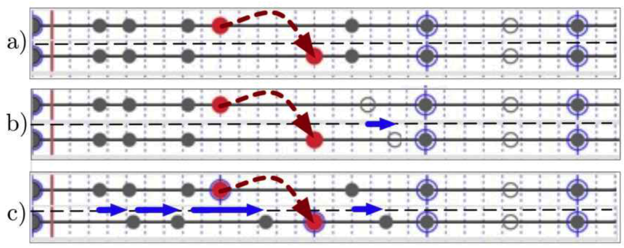


Figure 1. Impact of the time-modification of a timed-item (horizontal arrows indicate the changes yield by the initial item translation). a) Simple translation of a *timed* item (with timed neighbours — no side-effect). b) Translation with update of an item with no explicit date on the right (preserving a constant ratio). c) Translation and time stretching between *master* timed items.

2.4 Rendering

The TIMED-SEQUENCE model also allows for a seamless integration of musical objects in a generic rendering system [9]. In addition to the previous API, the method **collect-actions** allows a TIMED-SEQUENCE to return a list of actions and tasks to execute, related to the rendering of the structure in a given time interval. Actions can just be assigned to TIMED-ITEMs, so that playing the musical sequence can be modeled as reading through the list of TIMED-ITEMs in the corresponding interval, and triggering the corresponding actions. They can also be generated or interpolated to produce more complex sequences of actions dynamically: sampling can be achieved for instance by periodic calls to the **make-timed-item-at-time** function, not modifying the original sequence of TIMED-ITEMs.

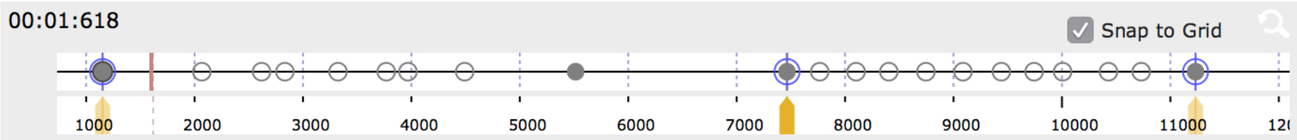


Figure 2. Main view of the *timeline-editor* associated to a TIME-SEQUENCE. The TIMED-ITEMs of different types are represented respectively as ○ (*untimed* / implicit time), ● (*timed*) and ⊙ (*master*).

3. GRAPHICAL EDITOR AND INTERACTIONS

A graphical user interface (GUI) is associated to the TIMED-SEQUENCE. This GUI allows the visualization of the TIMED-ITEMs and the execution of the most common operations on time structures. The main view (called *timeline-editor*, see Figure 2) lets the user select, add or delete timed items via calls to the API functions. For instance, adding a point in the *timeline-editor* (*command+click*) triggers a call to **make-timed-item-at-time** and adds the returned item to the sequence using **set-timed-item-list**. Setting/changing the date of TIMED-ITEMs is essentially done by translations on the timeline using the mouse. Keyboard short-cuts let user change the "type" of the items, mainly to turn standard items to master items and vice-versa.

Each master item creates a marker in the ruler visible at the bottom of the view. The marker, displayed as a yellow arrow, can be used as a proxy to move and synchronize the master item with the ruler. A cursor is also displayed as a vertical red line (around 1600 ms in Figure 2), which can be linked to the playback and rendering functionalities of the host environment to display or set the current play-time.

4. IMPLEMENTATION OF MUSICAL OBJECTS

A core set of objects can be defined in the generic framework introduced so far. It is possible to build such objects directly on top of the proposed architecture (Section 4.1), or by adapting existing objects to implement its API (Sections 4.2 and 4.3). The TIMED-SEQUENCE API and the *timeline-editor* handle the time representation and manipulations of the integrated objects.

4.1 Basic Timed Sequences

The TIMED-SEQUENCE model defines a generic type of object to use in computer music applications. We call DATA-STREAM a simple sequence of timed events directly inheriting from TIMED-SEQUENCE. The timed items in a DATA-STREAM are called DATA-FRAMEs. They contain data and perform specific actions to render these data. Subclasses of DATA-FRAME include for instance MIDI events or Open Sound Control (OSC) bundles, where data is a set of messages and the rendering transmits the messages via dedicated protocols.

Figure 3 shows a prototype editor created for DATA-STREAM objects, which plots DATA-FRAMEs (in this case, representing OSC bundles) along the horizontal time axis. The *timeline-editor* at the bottom is included as a component in the main editor, and enables time manipulations on the data frames, such as local/piecewise stretch and compression, snap to grid, etc.

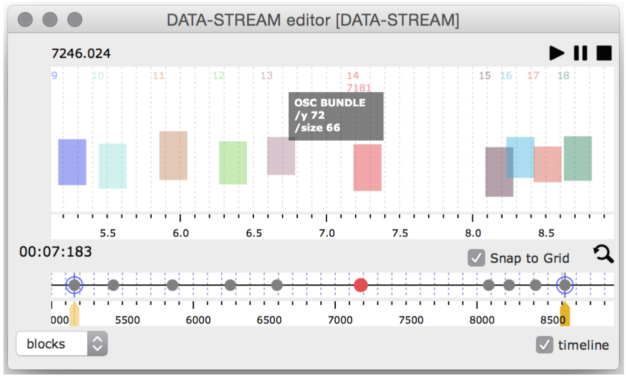


Figure 3. Editor for a sequence of data (class DATA-STREAM) containing DATA-FRAMEs (in this case, OSC bundles). Each DATA-FRAME is represented as a coloured shape. The graphical parameters (shape, colour, size or vertical position of the frames) are assigned using an arbitrary mapping with the data. Note the *timeline-editor* GUI component at the bottom.

A specialization of the previous structures allows for the implementation of a PIANO-ROLL representation as shown on Figure 4. The class MIDI-NOTE is a specific DATA-FRAME containing pitch, velocity and duration information. When rendered, it produces two actions sending MIDI *key-on* and *key-off* events.

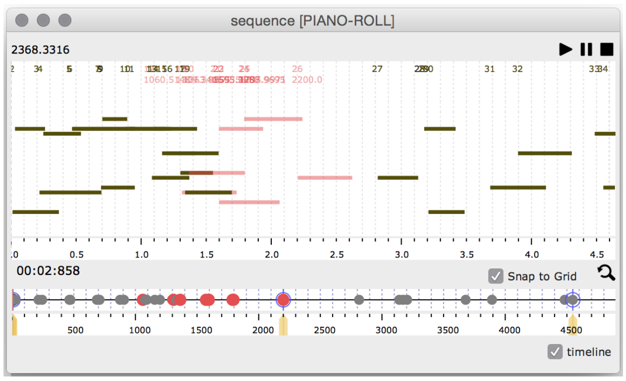


Figure 4. Editor for a sequence of MIDI notes (PIANO-ROLL) including a *timeline-editor*.

Since the model lets musical objects being built without exhaustive specific timing, one could consider specifying the date of two master items in a DATA-STREAM (DATA-FRAMEs or MIDI-NOTEs) and let the system interpolate intermediate time values.

4.2 Timed Controllers: Curves and Automations

The TIMED-SEQUENCE API can also be used to improve time manipulations in existing objects, such as the timed controllers and automations commonly used in computer music systems. It is straightforward to implement the accessors presented in Section 2 with a break-points function (BPF) object, which contains an ordered list of 2D points, and where the time dimension is implicitly associated to one of these dimensions (x / horizontal axis). Likewise, the *timeline-editor* can be embedded as a component in a BPF editor, as presented in Figure 5.

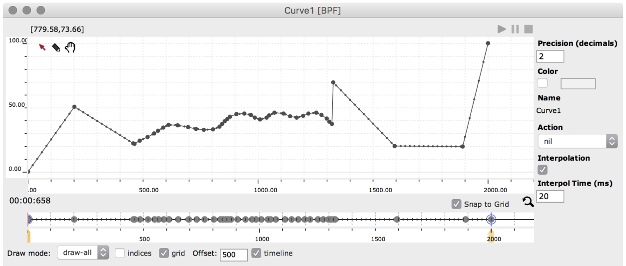


Figure 5. Editor of a break-point function implementing the TIME-SEQUENCE API, including a *timeline-editor* component. Each point in the graph is considered as a TIMED-ITEM.

4.3 Trajectories

As mentioned in the introduction, timing in trajectory specification is often a delicate problem. Indeed, although efficient interfaces exist to draw and design 2D or 3D curves, time specification must generally be done piecewise or using a global duration given to the entire movement.

The integration of the TIMED-SEQUENCE model in 2D or 3D curves is done through the definition of "timed points", an extension of 2D/3D points (including x, y and z coordinates) and subtype of TIMED-ITEM. The embedded API and graphical interface allow the user to easily perform time manipulations on such graphical structures.

In Figure 6 two master points control the global scaling and synchronization of a trajectory. A timed point with a defined date near the middle of the sequence splits the overall morphology into two parts with relatively equivalent durations. The rest of the points have no explicit date (their positioning in time will be computed on-demand whenever needed, according to the dates of the timed and master points).

5. SYNCHRONIZATION

Synchronization is one of the main time-domain operations performed in musical software, and is often identified as a key element of computer music systems [10]. Our framework facilitates the implementation of intuitive synchronization tools by connecting multiple timelines.

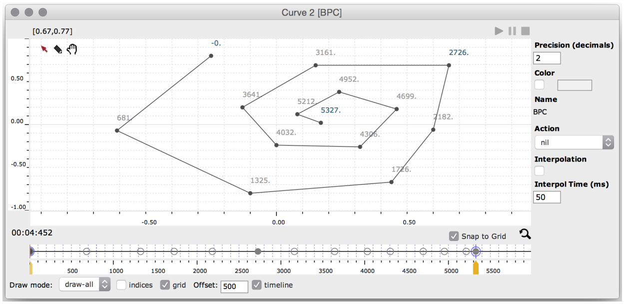


Figure 6. Editor of a 2D-curve in which only three points have an explicit date. The other (implicit) dates are deduced to preserve a constant speed across the different segments.

5.1 Internal Synchronization

Sticking with the sound spatialization example, let us now consider the case of multiple trajectories to control the motions of several sound sources. Time-synchronization strategies are crucial in this case: how to make specific (spatial) regions match in time?

The SPAT-SCENE is an interactive object/controller designed for sound spatialization processes, made up from a set of 3D trajectories (TIMED-SEQUENCEs) and connected to interactive (real-time) visualization and rendering [11]. Figure 7 shows the editor developed for this object, where the *timeline-editor* represents each trajectory in an individual track.

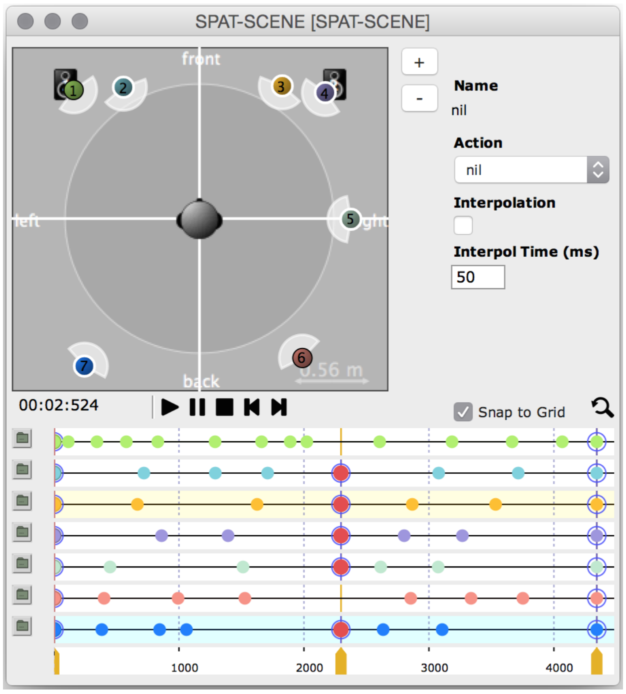


Figure 7. SPAT-SCENE editor. Each sound spatialization trajectory is associated with a timeline in the *timeline-editor* component at the bottom.

The aggregation of timeline views dedicated to each trajectory in a single *timeline-editor* enables the implementation of local synchronization strategies in such compound objects. First, the snap-to-grid functionality supports the adjustment of the time positions to the closest items or grid element within a certain range. This helps users to precisely set items from different sequences at the same date. Second, the markers created by the master items fuse when they are at the same date, and therefore act as proxies to all master items situated at this particular date. This allows several master items from different timelines that have been synchronized to be moved simultaneously, and facilitates stretching and compression operation to the neighbour segments of all corresponding objects. Figure 7 illustrates this behaviour with several master items being selected and moved through the user interacting with a single marker.

5.2 External Synchronization

The synchronization of master items can also occur at a higher structural level, when several temporal objects are combined to form more complex musical structures, for instance within a sequencer.

Figure 8 illustrates a scenario in which the user decides to synchronize two hand-drawn audio effect automations with specific parts of a sound file. The scenario is implemented in a musical container interface currently developed in OpenMusic on top of the same *timed-sequence* model.

Before synchronizing the objects, the composer annotates the sound file with two markers in order to define the beginning of two "sections" A and B, and adds a master point in *BPF2* to synchronize with these markers. All *master* items are collected out of the objects to their container's context, and lifted to the time ruler at the top of the sequencer interface (represented as vertical yellow lines). From there, the targeted operation requires dragging only a couple of markers in the time ruler.

In Figure 8.a the beginning of both automation curves (considered master points by default) is synchronized with the first marker in the sound (beginning of section A). The user action ① then synchronizes the central master point of *BPF2* (marker displayed in red on the ruler at the top) with the last point of *BPF1* (also considered as a master/synchronization point by default). This action ties together the two items, which can then be manipulated as a single entity.[2] In Figure 8.b, the user performs action ② to position this "grouped" marker at the beginning of section B of the sound file, thereby modifying the length of *BPF1* and the relative lengths of the two segments in *BPF2*. Finally, in Figure 8.c, action ③ connects the end point of *BPF2* to the end of the sound file, in order to adjust the duration of the second segment of the automation. Figure 8.d shows the resulting sequence with synchronized markers and items.

[2] Similarly to the internal synchronization, a *snap* functionality allows any dragged marker to be adapted to the position of the closest one within a given time window.

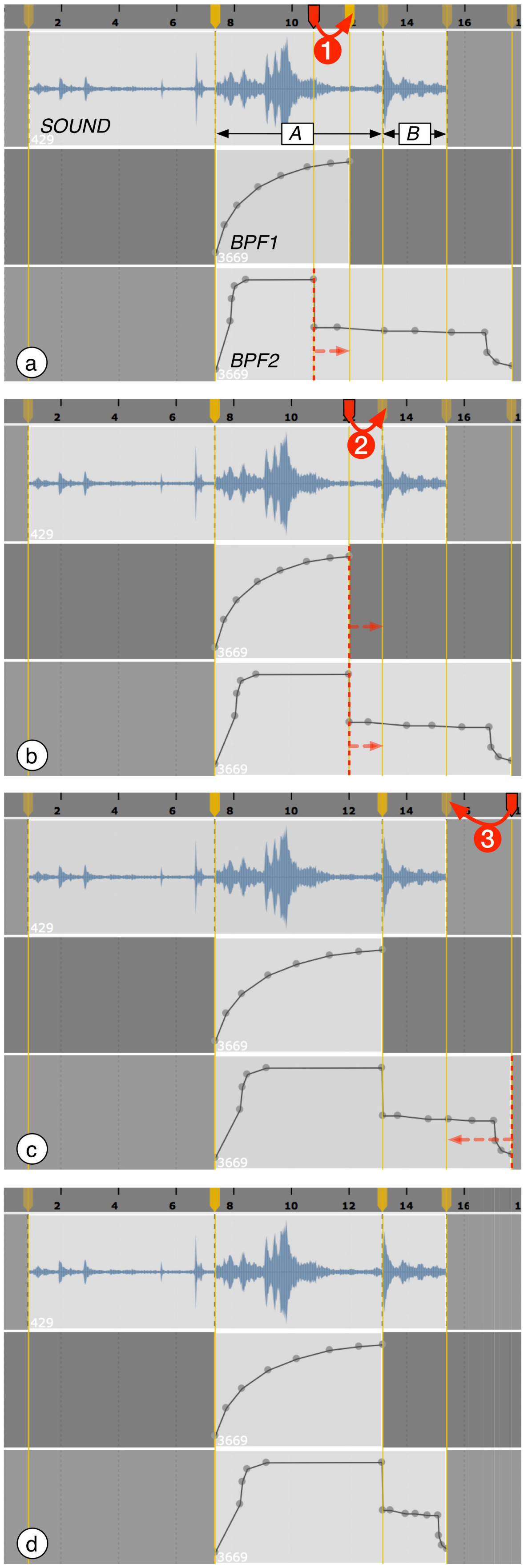


Figure 8. Synchronization of two effect automation curves with a sound. ①, ② and ③ represent user actions.

6. CONCLUSION

We introduced a programming and a graphical user interface framework for the representation of time in musical objects. We described the underlying concept of TIMED-SEQUENCE, an abstract representation containing an ordered set of TIMED-ITEMs, and the corresponding API used to represent musical objects through these simple structures. TIMED-ITEMs also facilitate explicit structuring of an object by defining temporal anchors that can be used to perform stretching and synchronization operations, both internally within an object, or externally with other objects.

This framework therefore enables expressive means to work with time either algorithmically or via graphical user interfaces, and provides end-users with consistent visualization and interaction mechanisms.

The TIMED-SEQUENCE model is currently implemented in the Common Lisp Object System [12] and used as a basis for the design of new interfaces and time structures in the OpenMusic environment.

Future work will focus on the representation of symbolic notation with our model in order to propose a comprehensive systems for composers to describe and process time structures in score-oriented frameworks. We also plan to explore more advanced interaction mechanisms, for instance considering weighted timed-items to control more sophisticated stretching and synchronization operations.

7. ACKNOWLEDGEMENTS

This work is supported by the French National Research Agency project EFFICACe ANR-13-JS02-0004.

8. REFERENCES

[1] R. B. Dannenberg, "Music Representation Issues, Techniques, and Systems," *Computer Music Journal*, vol. 17, no. 3, pp. 20–30, 1993.

[2] H. Honing, "Issues on the Representation of Time and Structure in Music," *Contemporary Music Review*, vol. 9, no. 1-2, pp. 221–238, 1993.

[3] J. Bresson and C. Agon, "Scores, Programs and Time Representations: The Sheet Object in OpenMusic," *Computer Music Journal*, vol. 32, no. 4, 2008.

[4] A. Allombert, M. Desainte-Catherine, J. Laralde, and G. Assayag, "A System of Interactive Scores Based on Qualitative and Quantitative Temporal Constraints," in *ARTECH 2008. Proceedings of the 4th International Conference on Digital Arts*, Porto, Portugal, 2008.

[5] M. Stroppa and J. Duthen, "Une représentation de structures temporelles par synchronisation de pivots," in *Colloque Musique et Assistance Informatique*, Marseille, France, 1990.

[6] B. Bel, "Time-setting of sound-objects: a constraint-satisfaction approach," in *International Workshop on Sonic Representation and Transform*, Trieste, Italy, 1992.

[7] J. Bresson, C. Agon, and G. Assayag, "OpenMusic: Visual Programming Environment for Music Composition, Analysis and Research," in *Proceedings of the ACM international conference on Multimedia – Open-Source Software Competition*, Scottsdale, USA, 2011.

[8] J. Bresson and M. Schumacher, "Representation and Interchange of Sound Spatialization Data for Compositional Applications," in *Proceedings of the International Computer Music Conference*, Huddersfield, UK, 2011.

[9] D. Bouche and J. Bresson, "Planning and Scheduling Actions in a Computer-Aided Music Composition System," in *Proceedings of the 9th International Scheduling and Planning Applications Workshop (SPARK)*, Jerusalem, Israel, 2015.

[10] P. Desain and H. Honing, "Towards a Calculus for Expressive Timing in Music," *Computers in Music Research*, vol. 3, pp. 43–120, 1991.

[11] J. Garcia, T. Carpentier, and J. Bresson, "Interactive-Compositional Authoring of Sound Spatialization," *Journal of New Music Research*, vol. 46, no. 1, 2017.

[12] R. P. Gabriel, J. L. White, and D. G. Bobrow, "CLOS: Integration Object-oriented and Functional Programming," *Communications of the ACM*, vol. 34, no. 9, pp. 29–38, 1991.

THE HOUSE HARMONIC FILLER: INTERACTIVE EXPLORATION OF CHORD SEQUENCES BY MEANS OF AN INTUITIVE REPRESENTATION

Ángel Faraldo, Perfecto Herrera, Sergi Jordà
Music Technology Group, Universitat Pompeu Fabra
name.surname@upf.edu

ABSTRACT

In this paper we present an interactive two-dimensional representation of musical chord progressions, integrated into a computer program that generates house music harmonic loops in MIDI format, based on a user's input. Our aim is to encapsulate relevant tonal information and display it in ways that are easy to understand for novices and untrained musicians, facilitating the creative exploration of musical ideas. We briefly reference previous work on tonal visualisation and interaction, and introduce some measures of tonal properties from the literature. We then present our system and describe the two-dimensional harmonic map, before discussing its outcomes and shortcomings, pointing at future lines of research in the conclusions.

1. INTRODUCTION

Computers have become one of the epicentres of professional music making. This has not only lowered costs in production, but has also facilitated music makers to be in closer contact with –and in many cases, to take complete care of– all stages of the music production chain, including tinkering and brainstorming, composing, layering and editing, recording, mixing, mastering and eventually, performing. Moreover, computers potentially provide a new realm of possibilities to the amateur musician and the curious mind, inviting them to engage in musical creation in unprecedented ways, through a variety of educational tools and games, digital musical instruments and accessible digital audio workstations (DAW's).

In this paper, we introduce a two-dimensional representation of chord sequences, that allows users (especially novices and musicians without formal education) to easily develop intuitions about certain tonal properties, like modality and tonal tension. Our visualisation method is integrated into a simple computer program that creates harmonic loops in house music style, a popular subgenre under the umbrella of Electronic Dance Music. As it will become apparent, house music holds a number of properties that make it suitable for our study.

Our explanation unfolds as follows: in the next Section, we briefly present related work in the areas of pitch space representations, digital interactive systems for harmonic exploration, as well as some existing measures of tonal properties. Then, we proceed to describe our computer program in Section 3, with an emphasis on the two-dimensional interactive space proposed. We discuss its outcomes and limitations in Section 4, before concluding with a summary and pointing at directions for future work.

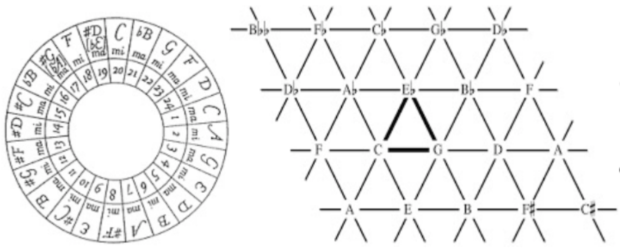


Figure 1. Typical renditions of pitch spaces: the circle of fifths (left) and a simple *Tonnetz* diagram (right).

2. BACKGROUND

2.1 Pitch Space Visualisations

Probably the most widespread representation of pitch spaces is the so-called circle of fifths, which represents relationships between adjacent keys. Richer in its representative power is Euler's *Tonnetz* (1739), displaying other intervalic relationships (major and minor thirds alongside the cycle of fifths), and upon which Riemann's influential tonal functional theory is grounded.

Several other pitch and chordal spaces have been developed since, following similar configurations. The geometrical representations of Longuet-Higgins (1962) and Balzano (1980), which attempt to represent harmonic distance, are worth mentioning, for they have been used in interactive musical systems [1, 2]. In any case, most of these abstractions almost invariably lead to lattice structures similar to the Tonnetz [3], with perhaps the exception of Shepard's [4] and Chew's [5] helicoidal models, that attempt to bring closer the pitch-class, chordal and key domains.

Recently, Bergstrom et al. [6] developed a system (*isochords*) that visualises chord progressions and voicings at playback time, resembling an animated Tonnetz. Other efforts towards visual analysis of tonal structure are mostly grounded in the works of Sapp [7, 8], evolving into interesting analysis methods using Self Organising Maps [9, 10] and simultaneously addressing multiple temporal scales [11]. However, these representational tools are developed with the analyst in mind, rather than the music creative, and cannot be used in real time.

2.2 Measuring Harmonic Properties

Most of the representational methods we just outlined were created to give account of certain tonal properties, be these in the realms of purely music-theoretical concepts, like the Tonnetz or the Spiral Array [5], or in the context of music cognition, attempting to illuminate the ways in which we humans listen to music [3, 4, 12]. It is in this area where a number of measures of perceptual distance have arisen, such as Lerdahl's distance indicator between chords in the context of multiple musical keys [3]. Recently, Bernardes et al. [13] introduced a novel measurement that estimates the perceptual proximity and consonance of note aggregates based on a 12-dimensional Tonal Interval Space, and which they use in their own generative system.

In the domain of harmonic consonance, Parncutt, has devoted a monograph to the study of the perceptual consonance of chords and sequences [14], after the pioneering works of Terhardt [15, 16]. However, this line of work is inevitably connected to the study of timbre and psychoacoustics, and lays slightly off our discussion.

In the field of music theory, Temperley [17] suggested different operations to measure various tonal properties of pitch-class sets (henceforth pc-sets). Grounded on Euro-classical [1] music theory and corpus analysis, he designed a Bayesian framework to measure the *tonal implication* (the key that a given pc-set implies), *tonal ambiguity* (a measure of the ambiguity of a pc-set to suggest one or several keys) and *tonalness* (the degree to which a pc-set is characteristic of the style he is studying) of pc-sets.

2.3 Interacting with Harmony

Several systems have been proposed to create, modify and more generally interact with harmonic spaces in digital environments, be these chord progressions or scales. A pioneering work in the field, Levitt's *Harmony Grid* (1986), lets the user hover with the mouse over different pitch space representations, sounding individual notes or various chord types (depending on the mode of operation) in response [1]. Soon after, Holland developed a number of educational programs about harmony in which the user could play simple chord progressions by tracing lines with a mouse over a Longuet-Higgins relational space [2]. Bernardes et al. [13] recently proposed a generative harmonic model based on a set of parameters (chord vocabulary, consonance factor and distance) selected by the user.

In the realm of bodily interaction, Gatzsche et al. [19] developed a system to physically interact with tonal spaces with hardware controllers, inspired in Chew's Spiral Array [5]. Similarly, Adeney came up with a multimedia environment in which a performer can literally step over different chord symbols projected on the floor as a two-dimensional grid, creating progressions based on tonal functional harmony as the performer moves [20]. Figure 2 shows screenshots of the Graphical User Interfaces (GUI's) of the systems we just mentioned.

[1] We take this term from Tagg [18] to refer to European Classical Music of the so-called common practice repertoire, on which most treatises on harmony are based.

3. THE HOUSE HARMONIC FILLER

In this section, we describe our tentative model for interacting with chord progressions in real time. We provide it as an open source program, written in Pure Data [2] and available online. [3] The program reads chord sequences from MIDI files, analyses their harmonic content and promotes simple variations like changing the voicing, register, inversion and rhythm of the sequences, generating MIDI data that can be sent out to any chosen device or DAW.

We refer to the prototype we are describing as the *House Harmonic Filler*, inspired by Moore's nomenclature [21]. According to Moore, popular music styles can be differentiated and characterized by observing four basic textural functional layers, namely the explicit beat layer, the functional bass layer, the melodic layer and the harmonic filler layer.

3.1 Rationale

We are interested in assistive tools for creating electronic popular music based on corpus analysis. We believe this approach can help us overcome certain musical assumptions that might not apply to the modes of musical production under consideration (cfr. [22, 23] for a description of tonal properties of EDM), and help the novice or amateur electronic music producer to become familiar with the main features of a given style of music. Furthermore, statistical music analysis can be useful for the musicologist in the task of observing and formalising new operational principles.

Despite harmony not being a prominent aspect of many electronic popular music genres, it is still prevalent in those evolving directly from a song tradition, such as electro-pop and disco variants. Our choice to develop this research on a corpus of house music, is based on the following premises:

- House music is composed and performed mainly with digital technology.
- Its basic structural unit is the *loop*, normally a 2, 4 or 8-bar circularly repeating sequence that is usually layered together with other such loops, what creates clear harmonic units without cadential points.
- House music, especially so-called *deep house*, is usually composed with chord loops borrowed from styles such as soul, rhythm-and-blues or even jazz, using extended chords other than simple triads, what makes it interesting in purely harmonic terms.
- Regarding instrumentation, deep house tracks often present acoustic instruments, such as pianos, vibraphones as well as an extensive use of vocals, what would eventually let us compare the output of our system with corpora of other popular music styles.

[2] `http://puredata.info/`
[3] `www.github.com/giantSteps/house-harmonic-filler`

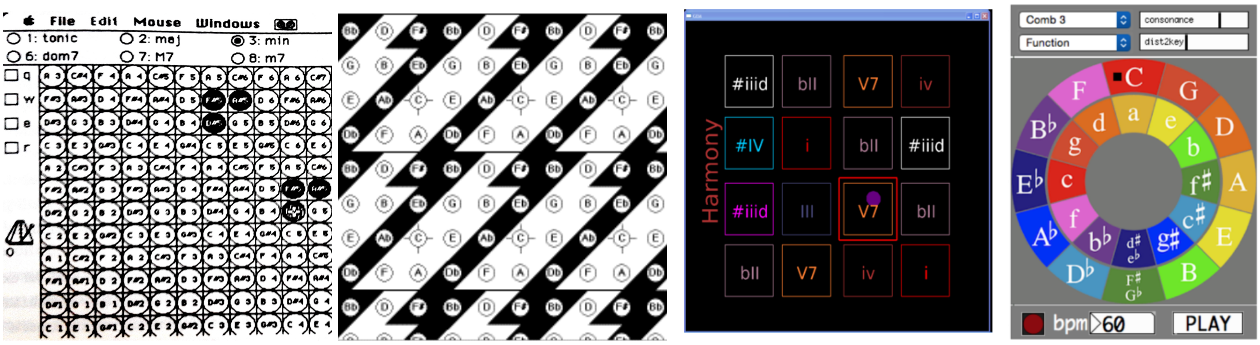


Figure 2. Graphical User Interfaces of various interactive applications dealing with harmony. From left to right: Levitt's *Harmony Grid* [1], Holland's *Harmony Space* [2], Adeney's *HarmonyGrid* [20], and Bernardes's *Conchord* [13].

3.2 Chord Shuttles and Loops

House harmonic loops normally consist of sequences of 2 or 4 bars, with a tendency to have a single chord per measure. However, 8-bar loops are less frequent, and in most cases, they result from a repetition of a 4-bar pattern with a small variation toward the end of the second half.

Currently, the House Harmonic Filler operates with two-bar shuttles and four-bar loops. According to Tagg [18], a chord shuttle consists in an ongoing oscillation between two chords, normally of equal duration and importance, what most of the times makes difficult –if useful at all– to determine which chord is the tonic in a traditional sense, effect which is enforced by the endless looping mechanism. In fact, the tonic feel in loop-based music is mostly determined by non-tonal compositional aspects, specially the explicit beat layer and the functional bass layer mentioned above, as well as the structural arrangement of loops, which results in timbral and density changes at regular hypermetric intervals [24, 25].

Moore [26] has studied the relation of specific chord progressions and different musical genres, attempting to identify specific styles (pop, rock and soul) based on a number of harmonic patterns. We consider this a potentially fruitful approach at differentiating specific electronic popular music sub-genres.

For the current study, we have used limited resources publicly available on the internet. We have gathered a collection of MIDI files containing homophonic chord loops under tags of *deep house piano*,[4] *classic house piano*[5] and *deep house chords*,[6] obtaining a total of 48 loops which we considered sufficient to study the visualization and interaction aspects that we are presenting.

3.3 General Operation

When we load a corpus of MIDI loops in the program, the system detects the number of chords in each file, their type, inversion and position in the 4-bar structure. The estimation of chord types is achieved via a lookup procedure against intervalic patterns stored in a dictionary. Once the chord sequence is determined, the system finds the root of the first chord in the loop, and establishes it as the tonic. We proceed in this fashion supported by the evidence that in short and cyclical chord progressions –as the ones described here,– the most natural presumption of a tonal centre lays on the first chord [24], especially if, as we cited above, this is emphasised by non-tonal features, such as density and timbral changes on a strict hypermetrical regularity. This hypothesis is also supported by Tagg [18]. After the analysis, the chord sequences are transposed to pitch-class 0 (C), so that users can select the key of the progression disregarding the original key.

All files in the corpus are analysed separately in terms of harmony and rhythm. This way, users can combine all rhythmic patterns in the corpus with the available chord sequences. Any chosen loop is presented in a simple display (Figure 3), showing the original rhythmic pattern in light-gray and the chord progression in Roman numeral notation. In the current version, user manipulations are limited to changes in register, spread and inversion, as well as the selection of key and some rhythmic transformations. Overall, the interface provides the following parameter controls:

- The *pattern* slider allows the user to select among existing rhythmic patterns, ordered by density and syncopation complexity.
- A *density* control changes the number of events in the loop according to an agnostic density transformer [27], allowing to create rhythmic deviations from the original pattern, presented in dark-gray in the interface (Figure 3).
- The *legato* parameter sets the relative duration of the events. Setting it to its maximum will make chords last until the next attack.
- The *octave* fader transposes the chord progression up or down.
- The *spread* slider controls the openness or closeness degree of chords, i.e., their spread over different octaves,
- while the *register* control affects their inversion type, folding chords upwards or downwards.

[4] www.loopmasters.com/genres/50-Deep-House/products/3829-Deep-House-Piano

[5] www.loopmasters.com/products/461-Classic-House-Pianos

[6] deep-house-chords.com/

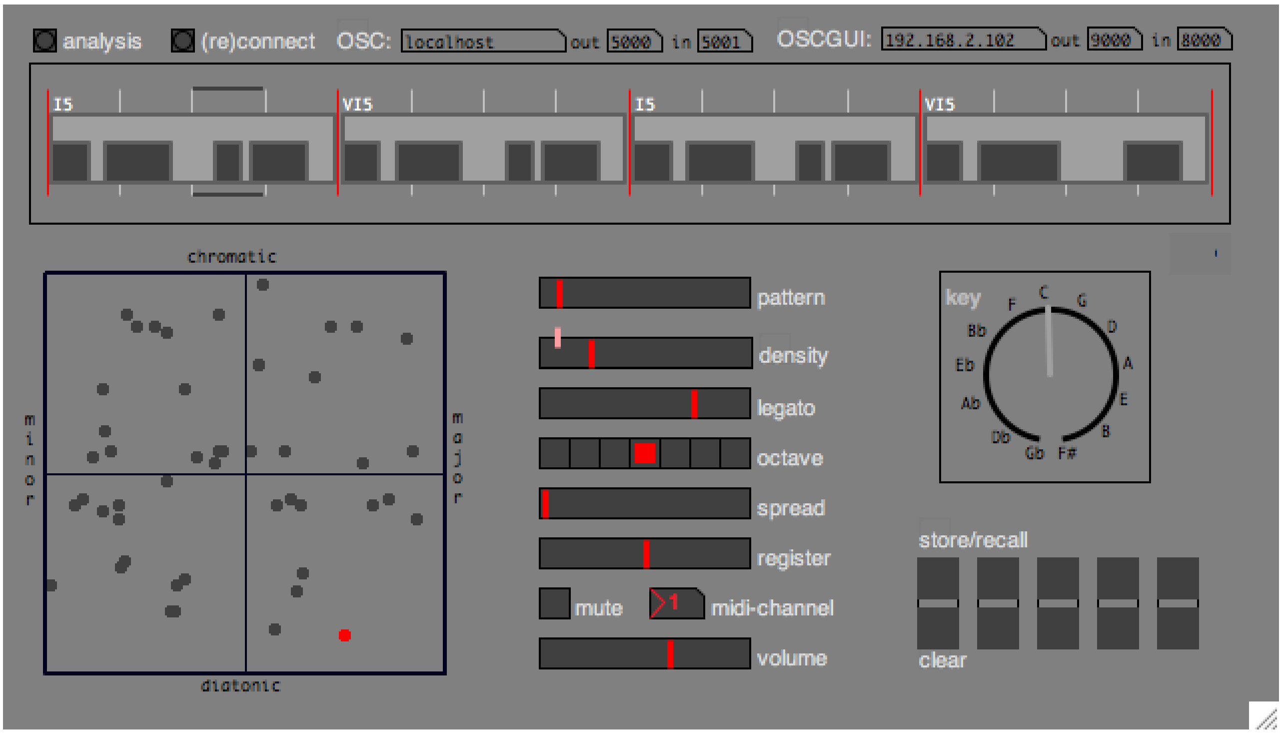


Figure 3. Graphical user interface of the House Harmonic Filler. It is composed of a chord progression selector (left), a simple visualisation of the chord sequence (top), and a few other controls to manipulate the sequence in real time.

- The *key* dial sets the key (tonic) of the progression. As we will explain in the next subsection, this is the pitch-class of the root of the first chord in the sequence. As the user might have observed in the GUI, the key control does not imply any specific modality. This is indeed only implied by the chord progression selected in the 2D space.

- Last, we provide a number of *memory slots*, so that users can store and recall between different states of the system, as well as regular *mute*, *midi channel* and *volume* controls to adjust the output parameters of the program.

3.4 The Harmony Map

3.4.1 Harmonic Analysis

Previous versions of the House Harmonic Filler had a one-dimensional drop-down menu, in which all harmonic progressions available were listed –after analysis,– in *chromatic* Roman numeral notation. All scale degrees are therefore expressed as chromatic intervals with the tonic. Figure 4 shows an example of a few such entries among which the user would need to choose:

`|I |♭VIImaj7 |Im9 |IIm9 |`

`|I9 |♭VIImaj9 |Vm11 |♭IIm9 |`

`|Im |VIIm7 |VIm7 |Vmaj7 |`

Figure 4. List of several chord progressions from our corpus of deep house loops.

A flat (♭) preceding a Roman numeral indicates a minor or diminished interval, and a lower case `m` after represents a minor chord; all non-flattened intervals refer to major or perfect intervals. This way, we intend to overcome the limitations of a mutually exclusive binary modal system, in which Roman numerals represent diatonic degrees of the scale. For example, in diatonic notation, `III` refers to the minor triad located one major third (3M) above a **major** tonic (`I`); and to the major triad one minor third (3m) above on a **minor** mode (`Im`). In chromatic notation, degrees refer unequivocally to a tonic, independently of the modality of the excerpt. Let us consider the following sequence:

`|Im9 |IIIm7 |V7 |♭VII7 |`

Reading this sequence in chromatic Roman numeral notation, if we assume, for example, `I` to be **A**, the progression results in:

`|Am9 |C♯m7 |E7 |G7 |`

The familiar reader will notice that this progression presents some mixed modality: first and last chords suggest an aeolian minor quality; however, this is broken by the second chord (which seems borrowed from the parallel key of **A major**); the third chord is ambiguous in this regard, since it belongs to both **A major** and **A minor** (harmonic).

This type of notation might be helpful to understand some harmonic features of the music under consideration, a somewhat jazzy house music. Especially, that modal variants (mixolydian, prhyrgian, etc.) are much more frequent than in common-practice harmony, and that there is a certain hybridisation of the major and minor modes. However, truth is that a regular user might be a bit disoriented when choosing a harmonic progression from a list of such entries according to her musical expectations.

3.4.2 *Interactive Visualisation*

In the latest version of the House Harmonic Filler we substituted the drop-down list of harmonic options with the two-dimensional interactive space shown at the left of Figure 3. In this new *harmony map*, all 4-bar chord progressions are represented as single dots in the space that users can click on to select them.

The harmony map intends to represent in a simple way some tonal properties of the chord progressions, regarding modality and tonal tension, over which the user can make herself an idea of the general tonal quality of the sequences without dealing with theoretical notations. Dots at the bottom of the graph are harmonically simpler than those at the top, whereas the horizontal axis represents a modal continuum from minor to major modes, passing through various modal variants.

In our grid, the x-axis represents, from left to right, a discrete progression of six possible modal variants, three minor and three major modes, arranged in the following order: *phrygian, aeolian, dorian, mixolydian, ionian* and *lydian*. The criterion for choosing this order was to arrange the various scalar possibilities from minor to major in the smoothest possible way, that is, keeping as many common notes as possible between nearby modes. In this setting, the typical modes of reference, *aeolian* (minor), and *ionian* (major) are located symmetrically at both sides. Figure 5 shows the modal arrangement of the x-axis indicating the changing notes between modes. To find the horizontal position of a given chord progression, we extract its pitch-class profile and calculate its euclidean distance to a set of stored templates with the modal variants, selecting the shortest interval.

The y-axis, also called the *diatonic-chromatic* axis, represents a measure of the overall tonal tension of the chord progression. We have obtained this measure by counting the number of different pitch-classes in the loop, together with the number of semitones between elements of the pc-set. This simple measure positions diatonic chordal progressions (with simple chords and diatonic notes) below, and brings the more colourful, *jazzy* or chromatic sequences to the upper part of the graph.

4. DISCUSSION

The grid in Figure 3 shows our corpus of house music distributed in the harmonic map. According to the representation, the corpus has a number of minor and major progressions (rich in modal variants), with sequences ranging from very diatonic at the bottom of the space to relatively chromatic ones. For example, the sequence corresponding to the red dot in Figure 3 corresponds to the following progression:

```
|I5 |VI5 |
```

Pr	Ae	Do	Mi	Io	Ly
b7	b7	b7	b7 →	7	7
b6	b6 →	6	6	6	6
5	5	5	5	5	5
4	4	4	4	4 →	#4
b3	b3	b3 →	3	3	3
b2 →	2	2	2	2	2
1	1	1	1	1	1
Pr	Ae	Do	Mi	Io	Ly

Figure 5. Modal distribution along the x-axis in the House Harmonic Filler. Degrees in red highlight interval changes across different modes.

The sequence is indeed highly diatonic. It presents a tonic chord without a third (`I5`) followed by the same type on the submediant (`VI5`). Therefore this progression has no semitones. If we translate all notes in the sequence to pitch-classes, reading the progression in **C**, we obtain the collection `[0,4,7,9]`. Although this pc-set seems clearly major (due to the presence of the tonic major chord in the set) it lacks the seventh degree –that would determine if this sequence is a ionian or mixolydian modal variant,– as well as the fourth –what would define it as ionian or lydian. Therefore, the dot is located in the middle of the major left half of the map, according to the distribution in Figure 5. This limitation to indicate the modal ambiguity of some chord progressions, is one of the main shortcomings of our system as it is.

Figure 6 shows images of the harmony map with two other small corpora. For the sake of comparison, we have created two sets of 20 MIDI files with 4-bar sequences of jazz standards (as notated in the Real Book [28]) and pop-rock songs from the Billboard dataset, respectively [29]. The only criterion to choose the sequences was that original chord progressions were repeated identically at least two times, in order to recreate the looping nature of our sequences, even if these types of music are not fundamentally based on loops. As it can be seen in Figure 6, the jazz corpus has slightly more presence in the two upper quadrants, what resonates with intuitions about jazz music being more chromatic (with chord extensions and local chromatic substitutions) that other musical styles. Alternatively, the grid representing items from the Billboard hits, clusters almost exclusively onto the lower quadrants, what again, aligns with regular intuitions about pop-rock music being mainly diatonic. Interestingly enough, a group of 5 points concentrate around the same area, exactly in the middle of the horizontal axis. That might be explained by the fact that a lot of rock music seems to be composed with very similar –if not identical– harmonic structures.

At the time of this report, we have conducted informal

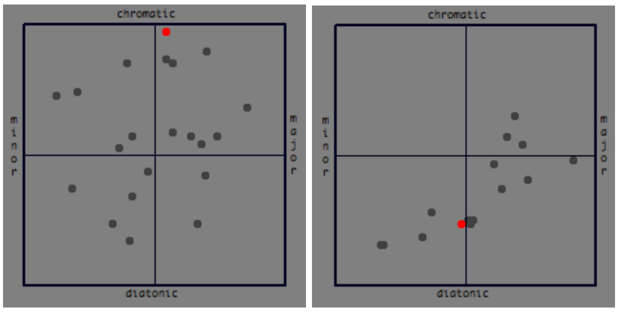


Figure 6. The two-dimensional harmony map with a small corpus of jazz progressions (left) and pop-rock music hits from the American Billboard (right).

testing with 15 subjects. All of them had some experience in music making, although only 5 of them were formally trained. In general, opinions were favourable regarding the use of the two-dimensional space. Untrained musicians regarded the axis labels to be sufficiently clear (major/minor, diatonic/chromatic), although they admitted they could not understand the Roman numeral notation in the progression display. Small frustrations arose from the limitation of the available progressions and variation strategies, one of the clear shortcomings of the program as it is now. At the moment, no chord expansion or substitution is possible, so the user is left with a few options affecting chordal voicings, plus a range of choice exclusively dependent on the preloaded corpus.

5. CONCLUSIONS AND FUTURE WORK

In this paper we presented the House Harmonic Filler, a computer program that generates house music loops and variations, with an emphasis on a two-dimensional interactive space that represents chord progressions in an intuitive way. The reduction of the chord sequences as dots was achieved with an *ad hoc* measure compressing relevant information on modality and diatonicism. However, certain modal ambiguities are not yet well represented in the map, something that we will like to address in future versions.

According to preliminary tests, this visualisation helps users unfamiliar with music theoretical notions to navigate among a closed set of possibilities. It is an endeavour for the future to design and carry on a systematic evaluation, based on user task-oriented experiments.

Similarly, we shall explore representations built over different measures, such as other types of harmonic distance (cfr. Lerdahl [3]), consonance ratios (e.g. Bernardes et al. [13]) and psychoacoustical dissimilarities, based on the measures by Parncutt [14].

We also envision an expansion of the program to allow interpolation between different chord progressions, overcoming some limitations that arise when only small corpora are available, as well as enabling a wider range of variations of the existing chord sequences. We see that an interpolation model could actually open up a space to think about chord substitutions and chordal expansions in ways that compel with the visual metaphor that we presented, as well as with musical intuitions and theoretical knowledge.

Acknowledgments

This research has been partially supported by the EU-funded *GiantSteps* project (FP7-ICT-2013-10 grant agreement number 610591).

6. REFERENCES

[1] D. Levitt, "Representing musical relationships in the harmony grid," in *Multimedia Interface Design in Education*. Springer, 1992, pp. 117–132.

[2] S. Holland, "Interface design for empowerment: A case study from music," in *Multimedia Interface Design in Education*. Springer, 1992, pp. 177–194.

[3] F. Lerdahl, *Tonal Pitch Space*. Oxford University Press, 2001.

[4] R. Shepard, "Structural representations of musical pitch," *The psychology of music*, pp. 343–390, 1982.

[5] E. Chew, "Towards a mathematical model of tonality," Ph.D. dissertation, Massachusetts Institute of Technology, 2000.

[6] T. Bergstrom, K. Karahalios, and J. C. Hart, "Isochords: Visualizing Structure in Music," in *ACM Proceedings of Graphics Interface*, 2007, pp. 297–304.

[7] C. S. Sapp, "Harmonic visualizations of tonal music," in *Proceedings of the International Computer Music Conference*.

[8] ——, "Visual Hierarchical Key Analysis," *ACM Computers in Entertainment*, vol. 4(4), pp. 1–19, 2005.

[9] P. Janata, "Navigating Tonal Space," *Computing in Musicology*, vol. 15, no. Tonal Theory for the Digital Age, pp. 39–50, 2007.

[10] P. Toiviainen, "Visualization of Tonal Content in the Symbolic and Audio Domains," *Computing in Musicology*, vol. 15, no. Tonal Theory for the Digital Age, pp. 187–199, 2007.

[11] A. Martorell, "Modelling Tonal Context Dynamics by Temporal Multi-Scale Analysis," Ph.D. dissertation, Universitat Pompeu Fabra, 2013.

[12] C. L. Krumhansl, *Cognitive Foundations of Musical Pitch*. Oxford University Press, 2001.

[13] G. Bernardes, D. Cocharro, C. Guedes, and M. E. P. Davies, "Conchord: An Application for Generating Musical Harmony by Navigating in a Perceptually Motivated Tonal Interval Space," in *Proceedings of the International Symposium on Computer Music Multidisciplinary Research*, 2015, pp. 1–16.

[14] R. Parncutt, *Harmony: A Psychoacoustical Approach*. Springer-Verlag, 2014.

[15] E. Terhardt, "Pitch, consonance, and harmony," *Journal of the Acoustical Society of America*, vol. 55(5), pp. 1061–1069, 1974.

[16] "The Concept of Musical Consonance: A Link between Music and Psychoacoustics," *Music Perception: An Interdisciplinary Journal*, vol. 1(3), pp. 276–295, 1984.

[17] D. Temperley, "The Tonal Properties of Pitch-Class Sets: Tonal Implication, Tonal Ambiguity, and Tonalness," *Computing in Musicology*, vol. 15, no. Tonal Theory for the Digital Age, pp. 24–38, 2007.

[18] P. Tagg, *Everyday Tonality II: Towards a Tonal Theory of What Most People Hear*. The Mass Media Music Scholar's Press, 2014.

[19] G. Gatzsche, M. Mehnert, and C. Stöcklmeier, "Interaction with tonal pitch spaces," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2008, pp. 325–330.

[20] R. Adeney, "The HarmonyGrid : Music, space and performance in Grid Music Systems by," Ph.D. dissertation, Queensland University of Technology, 2011.

[21] A. F. Moore, *Song Means: Analysing and Interpreting Recorded Popular Song*. Ashgate Publishing, 2012.

[22] A. Faraldo, E. Gómez, S. Jordà, and P. Herrera, "Key Estimation in Electronic Dance Music," in *Proceedings of the 38th European Conference on Information Retrieval*, 2016, pp. 335–347.

[23] R. Wooller and A. R. Brown, "A framework for discussing tonality in electronic dance music," in *Proceedings Sound : Space - The Australasian Computer Music Conference*, 2008, pp. 91–95.

[24] K. Stephenson, *What to Listen for in Rock: A Stylistic Analysis*. Yale University Press, 2002.

[25] M. Spicer, "(Ac)cumulative Form in Pop-Rock Music," *Twentieth Century Music*, vol. 1(1), pp. 29–64, 2004.

[26] A. Moore, "Patterns of Harmony," *Popular Music*, vol. 11(1), pp. 73–106, 1992.

[27] S. Jordà, D. Gómez-Marín, Á. Faraldo, and P. Herrera, "Drumming with style: From user needs to a working prototype," in *16th International Conference on New Interfaces for Musical Expression*, 2016.

[28] *The Real Book – Volume I*. Hal Leonard, 1970.

[29] J. A. Burgoyne, J. Wild, and I. Fujinaga, "An expert ground truth set for audio chord recognition and music analysis." in *Proceedings of the 12th International Society for Music Information Retrieval Conference*, 2011, pp. 633–638.

GENERATING EQUIVALENT RHYTHMIC NOTATIONS BASED ON RHYTHM TREE LANGUAGES

Florent Jacquemard
INRIA Sorbonne Universités
STMS (IRCAM-CNRS-UPMC)
IRCAM, Paris, France
florent.jacquemard@inria.fr

Adrien Ycart
School of Electronic Engineering
and Computer Science
Queen Mary Univ. of London, UK
a.ycart@qmul.ac.uk

Masahiko Sakai
Graduate School of Information Science
Nagoya University, Japan
sakai@is.nagoya-u.ac.jp

ABSTRACT

We propose a compact presentation of languages of preferred rhythms notation as formal grammars. It is based on a standard structure of rhythm trees capturing a wide range of rhythms in Western notation.

As an application, we then describe a dynamic programming algorithm for the lazy enumeration of equivalent rhythm notations (i.e. notations defining the same durations), from the simplest to the most complex. This procedure, based on the notion of rhythm grammars has been implemented and may be useful in the context of automated music transcription and computer-assistance to composition.

1. INTRODUCTION

Music notation is for music very much like writing for language. It serves as a support to convey ideas, to keep track of them in time, and as a working support for musical expression. In natural languages, there can be several synonyms to designate the same entity (such as ”Rome”, ”Roma”, ”capital of Italy”, ”city of the seven hills”, ”eternal city”...) and one might prefer using one word over the others depending on the context, a special connotation, the linguistic register *etc*. Similarly, in common Western music notation, there are often many different ways to write a given sequence of durations, and the choice of one writing over another is up to the composer, and can be driven by many reasons, among which are the musical context in which it is written, or a particular interpretation or phrasing that the composer wants to imply.

This is an important problem in applications related to score generation, in particular automated music transcription [1], score editors, or composer assistance environments [2]. In this context, it is interesting to assist users as much as possible in choosing appropriate notations for what they want to express.

A first question that arises is the definition of the domain of rhythms notations that one want to consider: which divisions (tuplets) are allowed? at which level? how many levels of division can we nest? In other words, we need some formalism to describe *languages of rhythms*. This corresponds for instance to the *codebooks* in the transcription procedure of [3], which are defined by *subdivision schemas* (the sequence of authorized successive regular subdivisions of a time interval), or similar notions already found in former work on transcription [4, 5, 6]. Another example is the choices of quantization parameters in user preferences of music editors such as Finale.

A second question is the design of efficient algorithms for exploring the set of rhythm notations in a given language that satisfies some property such as, for instance, the set of rhythms defining the same effective sequence of durations.

In this work, we follow a language theoretic approach to address these problems of rhythm notation. We propose a notion of rhythm languages defined by formal context-free (CF) grammars, following [7] (Section 3). It generalizes the formalisms cited above, as well as a related formalism that we have used so far in a new tool for rhythm transcription [8]. The rhythms defined by such a grammar are, roughly, the derivation trees of the grammar, seen as rhythm trees (RT). The latter are a tree structure for the representation of timed sequences of events [9, 10], where the events are stored in the leaves of the tree and the duration are encoded in the tree structure, every branching defining a uniform division of a time interval (Section 2). Similar representations have been supported since many years as a native data structure for the representation of rhythms in visual environment for composition assistance based on functional programming languages such as Open Music or PWGL [2, 11].

Using standard formalisms such as CF grammars for the definition of rhythm languages allows to exploit efficient construction and decision procedures from the literature.

Here, we use an algorithm for the enumeration of RTs defined by a given acyclic grammar [12] (Section 3.4). The enumeration follows a rank assigned to RTs by weights added to the grammar's rules. These weights can be seen as a measure for rhythm complexity. The main advantages of this algorithm is that it does not need to compute all the RTs of the language in order to rank them, but instead, thanks to a monotony property of the weight functions, build the best trees from the best subtrees in a lazy way.

CF grammars are a concise and readable formalism to define RT languages, but they are also expressive. The languages they define are regular tree languages [13], and as such can be composed by Boolean operations. This en-

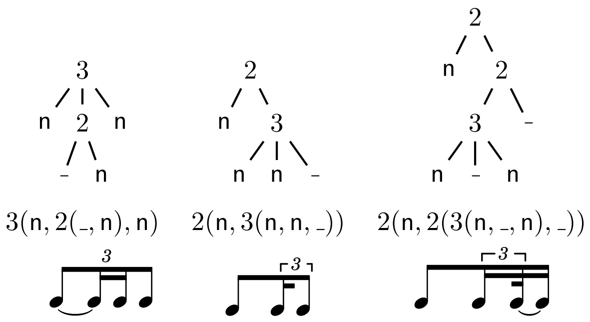


Figure 1. Rhythm Trees and corresponding notations.

ables incremental constructions of complex languages by composition of elementary languages, in a user-friendly fashion. In particular, it is possible to use Cartesian product in order to construct the intersection of two tree languages. We use this principle (Section 4) to construct a grammar defining the set of RTs from a given language L which correspond to a given sequence σ of duration values. Using the above-cited algorithm, we can then enumerate by ascending weight the RTs of L with duration value sequence σ. We apply the same principle to provide a way to find all rhythms that denote the merging of two voices into one (Section 5).

2. RHYTHM TREES

A rhythm tree (RT) is a hierarchical representation of a sequence of timed events, where the events are in the leaves, and the inner nodes define the durations by successive subdivisions of an initial time interval.

2.1 Syntax

In our settings, a RT is either:

a symbol n representing the beginning of an event, or

a symbol _ representing the continuation of an event, or

a tree of the form $t = p(t_1, \ldots, t_p)$ where p is a natural number (smaller than a given bound) and $t_1, \ldots, t_p$ is an ordered sequence of RTs.

The node labeled with p is called the *root* of t, and the sub-RTs $t_1, \ldots, t_p$ are called the *children* of t.

Note that every inner node of a RT is labelled with the arity (outer degree) of the node. We adopt this redundant notation only for readability purpose.

Every occurrence of the symbol n represents the starting date (onset) of an event. Some alternative symbols could be used for distinguishing different kinds of events: pitched notes, chords, rests *etc* (see Section 2.3). In this paper however, we always use n for the sake of simplicity.

The _ symbol represents a a note that is tied to the previous one. In some cases, those tied notes can be denoted by dots, see Section 2.3.

Example 1 *We present in Figure 1 three examples of RTs and the corresponding rhythm notations according to the duration semantics defined in Section 2.2.*

The first RT on the left is a triplet. Its second child is further divided in 2 parts, and the first part is a continuation. This means that the corresponding note (second note of the RT) is tied to the previous note (first note).

In the second RT, we have a division by 2 and then by 3, and a tie between the two last notes.

The last RT presents three levels of division, and two ties. Note that the depth in the RTs reflects the beaming level in the notation. For readability purposes, some tied notes have been merged, for instance in the second example, the last two tied sixteenth notes have been merged into one single eighth note.

2.2 Semantics

To every RT t, we associate a sequence of positive rational numbers called rhythmic value of t. This numbers correspond to the inter-onset intervals (IOIs) between the onset of events described in the RT. In the following, when considering the leaves of a tree, we will interchangeably use the term IOI instead of positive rational number.

In order to define the rhythmic value of the RT t, we associate to every node ν of t a positive rational number called *duration value*, and denoted by $dur(t, \nu)$.

Intuitively, the duration of a node is divided uniformly in the duration of its children, and _ is used to sum duration of leaves. Formally,

If ν is the root of t, then $dur(t, \nu) = 1$.

Otherwise, $dur(t, \nu) = \frac{dur(t, \nu_0)}{t(\nu_0)} + cdur(t, \nu)$, where

ν_0 is the parent of ν in t,

$t(\nu_0)$ is the label of ν_0 in t (*i.e.* its arity),

$cdur(t, \nu) = dur(t, \nu')$ if ν is a leaf and it has a next leaf ν' labelled with _ ,

$cdur(t, \nu) = 0$ otherwise.

The choice of duration 1 for the root node is arbitrary. It means that the fractions refer to divisions of one beat but with different root durations, they could as well refer to other orders of magnitude (bar *etc*). Note that the durations are expressed in beats (time units relative to a tempo), which makes the definition of a tempo useless here.

The events represented by a RT are stored in its leaves. Hence, the event's durations, *i.e.* the actual rhythm corresponding to a RT, denoted $val(t)$, is defined from the duration values of its leaves. More precisely, let $\nu_1 \ldots, \nu_k$ be the enumeration, in depth-first ordering, of the leaves of t labelled with n. The *rhythmic value* of t is the sequence

$$val(t) = dur(t, \nu_1), \ldots, dur(t, \nu_k).$$

It is the empty sequence when $k = 0$. Intuitively, if a rhythm tree t represents the notation of a rhythm, its rhythmic value represents the way this rhythm sounds.

Two RT t_1, t_2 are called *equivalent*, denoted $t_1 \equiv t_2$ iff $val(t_1) = val(t_2)$.

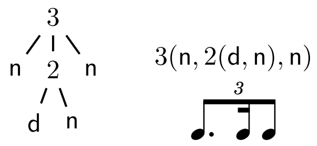


Figure 2. Alternative representation for dots.

Example 2 *The three RT of Figure 1 have a rhythmic value of* $[\frac{1}{2}, \frac{1}{6}, \frac{1}{3}]$ *and are therefore equivalent.*

Deciding the equivalence of two given RT t_1 and t_2 can be done simply by evaluating their respective rhythmic values (in linear time for each tree) and comparing the values.

2.3 Pitches, Rests, Grace Notes, Dots...

In the above RT representation, we consider only one generic symbol n to represent any kind of event. Straightforwardly, we could introduce new symbols labeling the leaves of RTs in order to encode any kind of finite information on events (pitches for notes or chords, trills, rests *etc*.

Alternatively, we could maintain along with a RT t a list of event's information of same length as the rhythm value of t. This list can be used later to recover this information (*e.g.* for rendering). These details are left out of this paper.

Grace notes are events of duration 0. The approach presented in this paper can deal with grace notes. Indeed, in a RT, one note preceded by one or several grace notes can be encoded in a single symbol, labelling a leaf. For instance, the symbol n_k could describe one note preceded by k grace notes. The definition of the rhythmic value of a RT should then be extended accordingly, by adding to the sequence k times the value 0 at the appropriate place.

Moreover, in some case we could use in RTs a dot symbol d in place of $\_$ the tie symbol, with the same semantics. For instance, in a RT containing a node n whose next sibling has the form $2(\_, \mathsf{n})$, the $\_$ could be replaced by the symbol d representing explicitly a dot notation, see Figure 2. The CF grammar formalism presented in Section 3 can do matching of patterns such as the previous one, hence it it is expressive enough to express the correct placement of dot symbols. In the rest of the paper, we voluntarily omit the dot symbols in grammars for simplicity purpose.

2.4 Related Rhythm Tree Formalisms

The definition of RTs used in this paper differs slightly from the earlier definitions of rhythm trees in Patchwork [9] and Open Music [10] (OMRT). In OMRTs, inner nodes are labelled by integer values denoting the relative length of durations of sibling nodes. For example, if an OMRT has 2 sons labelled with 1 and 2, the second son is twice as long as the first. Hence this tree corresponds to a triplet where the two last notes are tied. This is exactly equivalent as if the sons were labelled 2 and 4: only the ratios matter. Note that these numbers have nothing to do with the numbers used in the above RT encodings, which are just the arity of the inner nodes and carry no information. Therefore, processing OMRTs requires some integer arithmetics computations.

The standard formalism that we are using below to represent languages of rhythms (and the associated algorithms *e.g.* for language enumeration) can only handle finite sets of symbols. Therefore they are not appropriate to deal with OMRTs and we preferred here an all-symbolic encoding of RTs without integral values.

3. RHYTHM LANGUAGES

In this section, we propose a general finite representations for sets of RTs called rhythm grammars. Their purpose is to fix the kind of rhythm notations that we want to consider, using declarative rules.

For instance, one rule may express that at some level, division by three is allowed but division by five is not. One other rule can express that at some level we can have a leaf labeled with a note (n) or labeled with a continuation ($\_$).

3.1 Rhythm Grammars

A *rhythm grammar* (RG) is a context-free grammar $\mathcal{G} = (Q, q_0, R)$ where Q is a finite set of non-terminals, q_0 is the initial non-terminal and R is a finite set of production rules of one of the forms

$$q \to q_1, \ldots, q_p \text{ with } q, q_1, \ldots, q_p \in Q,$$

$$q \to a \text{ with } q \in Q \text{ and } a \in \{\mathsf{n}, \_\}.$$

Intuitively, these rules are applied from top to bottom to generate RTs by replacement of non-terminals by subtrees. A rule of the first kind (called *inner rule*) generate an inner node of a RT, expressing a division by p of the duration of this node. A rule of the second kind (called *terminal rule*) generate a leaf of a RT, and expresses that the label a is allowed at this leaf.

Generally in the literature, one considers the (context-free) language of words generated by CF grammars – in our case, they would be words over $\{\mathsf{n}, \_\}$. However, recall that RTs encode events in the leaves but also durations of these events in the inner nodes. Since we need these two informations for rhythm encoding, we need both kind of nodes, and therefore we will be more interested in the set of RTs generated by a RG (which is a regular tree language), than in the (context-free) language of the words found in a traversal of the leaves of these RT.

Formally, the language $\mathcal{L}_q(\mathcal{G})$ of a RG $\mathcal{G} = (Q, q_0, R)$ in non-terminal $q \in Q$ is defined recursively by

$$\mathcal{L}_q(\mathcal{G}) := \{a \text{ if } q \to a \in R\} \cup \bigcup_{q \to q_1, \ldots, q_p \in R} \{p(t_1, \ldots t_p) \mid t_1 \in \mathcal{L}_{q_1}(\mathcal{G}), \ldots, t_p \in \mathcal{L}_{q_p}(\mathcal{G})\}.$$

The language of $\mathcal{G}$ is $\mathcal{L}(\mathcal{G}) = \mathcal{L}_{q_0}(\mathcal{G})$.

Example 3 *Let us consider the following RT $\mathcal{G}$, with initial non-terminal q_0.*

$$\begin{array}{lll} q_0 \to \mathsf{n} & q_2 \to \_ & q_3 \to \_ \\ q_0 \to q_2, q_2 & q_2 \to \mathsf{n} & q_3 \to \mathsf{n} \\ q_0 \to q_3, q_3, q_3 & q_2 \to q_4, q_4 & q_3 \to q_5, q_5 \\ & q_2 \to q_5, q_5, q_5 & \\ q_4 \to \_ & q_5 \to \_ & \\ q_4 \to \mathsf{n} & q_5 \to \mathsf{n} & \\ q_4 \to q_5, q_5, q_5 & & \end{array}$$

The three rules with left-hand side q_0 express that the root of RT in $\mathcal{G}$'s language can be either a single event n, or a division by 2, with children in q_2, or a division by 3, with children in q_3.

At the next level, starting with q_2, we can have leaves labeled with $\_$ or n, or division by 2 (with children in q_4), or division by 3 (with children in q_5). Starting with q_3, we can have only leaves or division by 2 (with children in q_5). At the next level, q_5 will generate only leaves ($\_$ or n) whereas at q_4 we can have a last division by 3.

Note that rules of $\mathcal{G}$ are acyclic, and hence define a finite RT language.

All the three RTs of Figure 1 are in the language of this RG. The precise computations to obtain these RT are described in Example 5 below.

3.2 Weighted Rhythm Grammars

We extend RG into weighted rhythm grammars (WRG) by adding a weight (real value) to each production rule, with the notations $q \xrightarrow{w} q_1, \ldots, q_p$ and $q \xrightarrow{w'} a$ respectively for weighted inner and terminal rules with weights w and w'.

Example 4 *We describe below a weighted extension of the RG of Example 3.*

$$\begin{array}{lll} q_0 \xrightarrow{0.1} \mathsf{n} & q_2 \xrightarrow{0.2} \_ & q_3 \xrightarrow{0.2} \_ \\ q_0 \xrightarrow{0.35} q_2, q_2 & q_2 \xrightarrow{0.1} \mathsf{n} & q_3 \xrightarrow{0.1} \mathsf{n} \\ q_0 \xrightarrow{0.45} q_3, q_3, q_3 & q_2 \xrightarrow{0.5} q_4, q_4 & q_3 \xrightarrow{0.5} q_5, q_5 \\ & q_2 \xrightarrow{0.6} q_5, q_5, q_5 & \\ q_4 \xrightarrow{0.2} \_ & & \\ q_4 \xrightarrow{0.1} \mathsf{n} & q_5 \xrightarrow{0.2} \_ & \\ q_4 \xrightarrow{0.75} q_5, q_5, q_5 & q_5 \xrightarrow{0.1} \mathsf{n} & \end{array}$$

The weights can be thought as penalties for symbols or divisions. For instance, every note n induces a minimal penalty of 0.1 whereas a continuation $\_$ induces a penalty of 0.2, in order to penalize ties. At level 0 (root), a duplet induces a penalty of 0.35 whereas for triplet it is a bit bigger (0.45). The situation is similar at lower levels. See Section 3.3 for lengthier discussion on choosing weights.

The weights in WRG are used to rank the RTs (by ascending weight). In order to associate to every RT a unique weight by a WRG, we use the following notion of run. Intuitively, a run represents the sequence of application of the grammar's production rules in order to obtain a RT. Formally, a *run* of a WRG $\mathcal{G} = (Q, q_0, R)$ on a RT t is a relabelling of the nodes of t with production rules of R, such that for every inner node ν labeled with $q \xrightarrow{w} q_1, \ldots, q_p$, for every $1 \le i \le p$, the i^{th} children node of ν is labeled by a rule of R of left-hand side q_i.

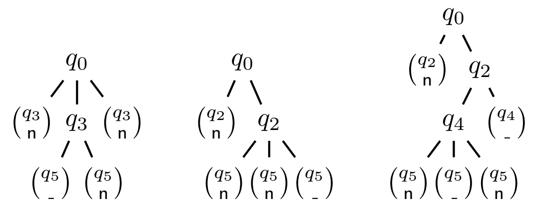


Figure 3. Runs on the RTs of Fig. 1 of the WRG of Ex. 4.

Example 5 *In Figure 3, we display runs of the WRG of Example 4 on the three RTs of Figure 1. For the sake of readability, we only display the left-hand side of rules on inner nodes and the non-terminal and symbol on leaves (the whole run can be easily recovered from this).*

Note that there can be several runs for one RT of the language (although it is not the case in the above example).

The *weight* of a run is the sum of weights of the rules at all nodes of this run. The weight associated to a RT t by a WRG $\mathcal{G}$ is either undefined if there exists no run of $\mathcal{G}$ on t, or the minimum of the weights of the runs of $\mathcal{G}$ on t.

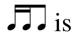
Example 6 *The weights associated by the WRG of Example 4 to the RTs of Figure 1 are respectively:*

$$\begin{array}{rl} 3(\mathsf{n}, 2(\_, \mathsf{n}), \mathsf{n}): & 1.45 \\ 2(\mathsf{n}, 3(\mathsf{n}, \mathsf{n}, \_)): & 1.45 \\ 2(\mathsf{n}, 2(3(\mathsf{n}, \_, \mathsf{n}), \_)): & 2.3 \end{array}$$

3.3 Choice of the Weight Values

The choice of the weights in production rules is crucial, as it is what will determine how the various RTs will be ranked. Generally speaking, we want RTs to be ranked according to their complexity. However, rhythmic complexity is difficult to define because it is highly subjective, and depends strongly of the context in which the rhythm is embedded. Some studies have tackled the perception of rhythmic complexity (see [14] for various definitions), but to the best of our knowledge, none have treated of the complexity of rhythm notation.

One naive definition would be to consider basic measures such as the size and the depth of the RT: the more nodes there are and the deeper the tree is, the more "complex" the rhythm is likely to be. The resulting weight function would be of the form : $q \xrightarrow{d+s*p} q_1, \ldots, q_p$ where d and s are two positive coefficients penalizing the depth and the size of the tree, respectively.

This definition shows its limits when comparing for example a quintolet and a sextolet. The quintolet is quite unusual, and will often be seen as complex, while a sextolet is much more common. Size alone cannot account for the complexity of a notation. Same with depth:  is generally considered less complex than a quintolet, even though its tree is deeper.

To go around this problem, a measure for notation complexity that tries to take into account these musical considerations was proposed in [8]. It is based on a ranking of divisions complexity proposed in [15], in which arities are ranked as follows, from less complex to more complex : 1, 2, 4, 3, 6, 8, 5, 7,... We can thus define a function $\beta(p)$ representing the penalty associated to an arity p, and the weight function becomes $q \xrightarrow{\beta(p)} q_1, \ldots, q_p$

Still, in this measure, the function $\beta(p)$ is arbitrarily determined, even though its contour is determined based on musical considerations. To determine it more relevantly, we could perform large-scale analysis of music score corpora, and choose $\beta(p)$ according to the frequency of each arity (the higher the frequency, the lower the weight). We could go even further in this corpus-based estimation of weights by assigning to each production rule of our RG a weight inversely proportional to its frequency in a corpus of scores. This would allow us to not only take into account the arity of the nodes in the RT, but also the depth at which they are found, and even the series of subdivisions they are in, relatively to the grammar chosen.

Another advantage of this corpus-based approach is that it can capture specific styles of notation. For example, if the weights were computed with a corpus of contemporary classical music scores, quintolet, septolet, and other artificial subdivisions would be less penalised than if a corpus of baroque music was used. It would even be possible for a composer to get personalised recommendations matching their own preferences of notation by using a corpus of their previous pieces.

3.4 k-best Parsing Algorithm

The k-best Parsing Algorithm [16] is a dynamic programming algorithm that, given a weighted context-free grammar $\mathcal{G} = (Q, q_0, R)$, enumerates its k best runs (ranked by their weight), where k is a parameter given by the user. In [8], we proposed an implementation of this algorithm from which the following stems. Here, we apply this algorithm to the WRG $\mathcal{G}$, and enumerate its least-weight RTs.

The algorithm uses a table which associates to every non-terminal $q \in Q$ two ordered lists of runs of $\mathcal{G}$:

$bests[q]$, containing the runs of $\mathcal{G}$ on RTs with q at the root and of minimal weight, as well as their weights.

$cands[q]$ (candidate bests), containing runs of $\mathcal{G}$ among which the next best will be chosen.

In each of those lists, the runs are not stored in-extenso. The elements of those lists are lists of pairs of the form $\big(\langle q_1, i_1\rangle, \ldots, \langle q_p, i_p\rangle\big)$, where every $q_j \in Q$ and every i_j is an index in $bests[q_j]$, and R contains a rule $q \xrightarrow{w} q_1, \ldots, q_p$. Those lists will also be called runs in what follows.

3.4.1 Initialization of the table

For each $q \in Q$, $bests[q]$ is initialy empty, and $cands[q]$ is initialized with one run $\big(\langle q_1, 1\rangle, \ldots, \langle q_p, 1\rangle\big)$ for each rule $q \xrightarrow{w} q_1, \ldots, q_p$ in R, with an unknown weight, and one run () of weight w' for each rule $q \xrightarrow{w'} a$ in R.

3.4.2 Algorithm

The algorithm evaluates the weights of candidates in the table in a lazy fashion, and transfers candidates of minimal weights in the best list. It works recursively, by computing the weight of each run from the weights of its sons. The main function, $best(k, q)$, returns the k-th best run of root q:

1. If $best[q]$ already contains k elements or more, then return the the k-th run of this list and its weight.

2. Otherwise, evaluate the weight of all runs in $cand[q]$ as follows: for a run $\big(\langle q_1, i_1\rangle, \ldots, \langle q_p, i_p\rangle\big) \in cand[q]$ whose weight is unknown, call recursively $best(i_j, q_j)$ for each $1 \leq j \leq p$, and then evaluate the weight by summing the weights of the sub-runs and the weight w of the rule $q \xrightarrow{w} q_1, \ldots, q_p$.

3. Once all the weights of the runs in $cand[q]$ have been evaluated, remove the run of smallest weight from this list, add it to $best[q]$ (together with its weight). Then add to $cand[q]$ the following next runs, with unknown weight: $\big(\langle q_1, i_1{+}1\rangle, \ldots, \langle q_p, i_p\rangle\big)$, $\big(\langle q_1, i_1\rangle, \langle q_2, i_2{+}1\rangle, \ldots, \langle q_p, i_p\rangle\big)$, $\ldots$, $\big(\langle q_1, i_1\rangle, \ldots, \langle q_p, i_p + 1\rangle\big)$. This step ensures us that the next best is in the candidate list. Repeat 2. and 3. until $best[q]$ or $cand[q]$ is empty (in the later case, $best(k, q)$ is undefined).

4. ENUMERATION OF EQUIVALENT RHYTHMS

Now we have all the elements to represent and enumerate the set of rhythms equivalent to a given rhythm. Let us first reformulate precisely, in the above settings, the problem we are interested in:

given a weighted rhythm grammar $\mathcal{G}$ and a non-empty sequence σ of positive rational numbers (IOIs),

return a weighted rhythm grammar $\mathcal{G}_\sigma$ such that $\mathcal{L}(\mathcal{G}_\sigma) = \{t \in \mathcal{L}(\mathcal{G}) \mid val(t) = \sigma\}$.

Hence, given a RT t, the WRG $\mathcal{G}_{val(t)}$ will represent the set of WRT of $\mathcal{G}$ equivalent to t. Moreover, using the algorithm of Section 3.4, we can enumerate this set.

4.1 Grammar Product Construction

Let $\mathcal{G} = (Q, q_0, R)$. The construction of $\mathcal{G}_\sigma$ works as a Cartesian product, following the similar construction for tree automata [13]. The non-terminals of $\mathcal{G}_\sigma$ are pairs of the form $\langle \tau, q\rangle$ where $q \in Q$ and τ is a part of σ, in a sense explained below. Its initial non-terminal is $\langle \sigma, q_0\rangle$, and every production rule of $\mathcal{G}_\sigma$ is either of the form: $\langle \tau, q\rangle \xrightarrow{w} a$ such that $q \xrightarrow{w} a \in R$ and τ is a singleton sequence, or $\langle \tau, q\rangle \xrightarrow{w} \langle \tau_1, q_1\rangle, \ldots, \langle \tau_p, q_p\rangle$ such that $q \xrightarrow{w} q_1, \ldots, q_p \in R$ and $\tau_1, \ldots, \tau_p$ is a partition of τ in p parts of equal length, where the length of a sequence of positive rational numbers is the sum of its elements.

The only tricky point for partitioning τ in p parts is that it may require to split some rational number r in two parts r_1 and r_2, such that $r = r_1 + r_2$, where r_1 will be the last element of some τ_i and r_2 will be the first element of τ_{i+1} (necessarily a continuation). For instance, the partition of

$[\frac{1}{2}, \frac{1}{6}, \frac{1}{3}]$ in two parts is $[\frac{1}{2}]$, $[\frac{1}{6}, \frac{1}{3}]$, but the partition of $[\frac{1}{6}, \frac{1}{3}]$ in two parts of equal length is $[\frac{1}{6}, \frac{1}{12}]$, $[\frac{1}{4}]$, and in this partition, $\frac{1}{4}$ has to be a continuation since the duration $\frac{1}{3}$ has been cut into $\frac{1}{12} + \frac{1}{4}$.

Let us state this precisely. We consider non-empty sequences of positive rational numbers with a sign: $-\tau$ means that the first element of the sequence τ is a continuation and $+\tau$ means that it is not (the sign $+$ may be omitted). Now we define the concatenation $\odot$ of signed sequences of positive rational numbers by:

$$\delta[r_1, \ldots, r_n] \odot +[s_1, \ldots, s_m] = \delta[r_1, \ldots, r_n, s_1, \ldots, s_m]$$
$$\delta[r_1, \ldots, r_n] \odot -[s_1, \ldots, s_m] = \delta[r_1, \ldots, r_n + s_1, s_2, \ldots, s_m]$$

where δ is $+$ or $-$ and $n, m \geq 1$.

The *length* of a signed sequence $\sigma = \delta[r_1, \ldots, r_n]$ of rational numbers is $\sum_{i=1}^{n} r_i$, denoted $\|\sigma\|$. A p-partition of a signed sequence τ (for $p > 0$) is a sequence $\tau_1, \ldots, \tau_p$ of signed sequences such that $\tau_1 \odot \ldots \odot \tau_p = \tau$ and $\|\tau_1\| = \ldots = \|\tau_p\| = \frac{\|\tau\|}{p}$. Note that it is unique for a given couple (τ, p).

Example 7 $+[\frac{1}{2}, \frac{1}{6}, \frac{1}{3}] = +[\frac{1}{2}] \odot +[\frac{1}{6}, \frac{1}{3}]$ *and* $+[\frac{1}{6}, \frac{1}{3}] = +[\frac{1}{6}, \frac{1}{12}] \odot -[\frac{1}{4}]$.
These two concatenation are 2*-partitions.*

Now we can describe precisely the construction of $\mathcal{G}_\sigma$ from the RG $\mathcal{G}$ and the sequence σ. Every non-terminal of $\mathcal{G}_\sigma$ is a pair made of a signed sequence τ of positive rational numbers and a non-terminal q of $\mathcal{G}$, denoted by $\langle \tau, q \rangle$.

1. $\mathcal{G}_\sigma$ contains the non-terminal $\langle +\sigma, q_0 \rangle$ (initial).

2. For every non-terminal $\langle \tau, q \rangle$ of $\mathcal{G}_\sigma$, and every production rule $q \xrightarrow{w} q_1, \ldots, q_p$ of $\mathcal{G}$, $\mathcal{G}_\sigma$ contains the production rule $\langle \tau, q \rangle \xrightarrow{w} \langle \tau_1, q_1 \rangle, \ldots, \langle \tau_p, q_p \rangle$ such that $\tau_1, \ldots, \tau_p$ is the p-partition of τ, and $\mathcal{G}_\sigma$ contains the non-terminals $\langle \tau_1, q_1 \rangle, \ldots, \langle \tau_p, q_p \rangle$.

3. For every singleton non-terminal $\langle +[r], q \rangle$ of $\mathcal{G}_\sigma$ ($r \in \mathbb{Q}_+$) and every production rule $q \xrightarrow{w} \mathsf{n}$ of $\mathcal{G}$, $\mathcal{G}_\sigma$ contains the production rule $\langle +[r], q \rangle \xrightarrow{w} \mathsf{n}$.

4. For every singleton non-terminal $\langle -[r], q \rangle$ of $\mathcal{G}_\sigma$ and every production rule $q \xrightarrow{w} \_$ of $\mathcal{G}$, $\mathcal{G}_\sigma$ contains the production rule $\langle -[r], q \rangle \xrightarrow{w} \_$.

The size of $\mathcal{G}_\sigma$ is at most the size of $\mathcal{G}$ times the size of σ. The correctness of construction follows from the property that: $\mathcal{L}_{\langle \tau, q \rangle} = \{t \in \mathcal{L}_q \mid val(t) = |\tau|\}$, where $|\tau|$ denotes the absolute value of τ, *i.e.* the sequence τ without its sign.

4.2 Examples

Example 8 *Let us consider the application of the above procedure to the WRG $\mathcal{G}$ of Example 4 and the rhythmic value $\sigma = [\frac{1}{2}, \frac{1}{6}, \frac{1}{3}]$.*

The initial non-terminal of $\mathcal{G}_\sigma$ is $\langle \sigma, q_0 \rangle$ (we omit the $+$ sign).

From the production rules of $\mathcal{G}$ starting with q_0, and 2*- and* 3*-partitions of σ, we obtain*

$$\langle \sigma, q_0 \rangle \xrightarrow{0.35} \langle [\tfrac{1}{2}], q_2 \rangle, \langle [\tfrac{1}{6}, \tfrac{1}{3}], q_2 \rangle$$
$$\langle \sigma, q_0 \rangle \xrightarrow{0.45} \langle [\tfrac{1}{3}], q_3 \rangle, \langle -[\tfrac{1}{6}, \tfrac{1}{6}], q_3 \rangle, \langle [\tfrac{1}{3}], q_3 \rangle$$

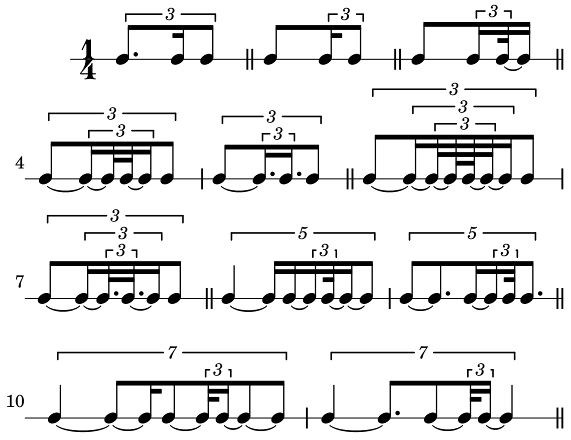


Figure 4. Enumeration for $\sigma = [\frac{1}{2}, \frac{1}{6}, \frac{1}{3}]$ and a more elaborate WRG than in Example 4. The RTs are separated by double bars. Single bar separate two version with or without dots of the same RT.

From the new non-terminals with singleton IOIs, we have

$$\langle [\tfrac{1}{2}], q_2 \rangle \xrightarrow{0.1} \mathsf{n} \qquad \langle [\tfrac{1}{3}], q_3 \rangle \xrightarrow{0.1} \mathsf{n}$$

From the new non-terminal $\langle [\frac{1}{6}, \frac{1}{3}], q_2 \rangle$, we have:

$$\langle [\tfrac{1}{6}, \tfrac{1}{3}], q_2 \rangle \xrightarrow{0.5} \langle [\tfrac{1}{6}, \tfrac{1}{12}], q_4 \rangle, \langle -[\tfrac{1}{4}], q_4 \rangle$$
$$\langle [\tfrac{1}{6}, \tfrac{1}{3}], q_2 \rangle \xrightarrow{0.6} \langle [\tfrac{1}{6}], q_5 \rangle, \langle [\tfrac{1}{6}], q_5 \rangle, \langle -[\tfrac{1}{6}], q_5 \rangle$$

and then:

$$\langle [\tfrac{1}{6}, \tfrac{1}{12}], q_4 \rangle \xrightarrow{0.75} \langle [\tfrac{1}{12}], q_5 \rangle, \langle -[\tfrac{1}{12}], q_5 \rangle, \langle [\tfrac{1}{12}], q_5 \rangle$$

From the other non-singleton non-terminal $\langle -[\frac{1}{6}, \frac{1}{6}], q_3 \rangle$ we have:

$$\langle -[\tfrac{1}{6}, \tfrac{1}{6}], q_3 \rangle \xrightarrow{0.5} \langle -[\tfrac{1}{6}], q_5 \rangle, \langle [\tfrac{1}{6}], q_5 \rangle$$

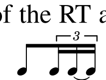
Finally, from the remaining singleton non-terminals:

$$\langle [\tfrac{1}{12}], q_5 \rangle \xrightarrow{0.1} \mathsf{n} \qquad \langle -[\tfrac{1}{12}], q_5 \rangle \xrightarrow{0.2} \_$$
$$\langle -[\tfrac{1}{4}], q_4 \rangle \xrightarrow{0.2} \_ \qquad \langle [\tfrac{1}{6}], q_5 \rangle \xrightarrow{0.1} \mathsf{n}$$
$$\langle -[\tfrac{1}{6}], q_5 \rangle \xrightarrow{0.2} \_$$

The language of this grammar contains the three RTs displayed in Figure 1, with the weights listed in Example 6.

Example 9 *With a more elaborate WRG $\mathcal{G}$ (10 non-terminals and 55 production rules) and the same σ as in Example 8, we obtain a WRG $\mathcal{G}_\sigma$ with 73 non-terminals and 79 production rules, whose language contains the 7 RTs displayed Figure 4 (with or without dots).*

One can notice in the examples that some of the notations proposed do not seem to be generated by the grammars described. For instance, in the middle example in Figure 1, the notation displayed does not match exactly the structure of the RT above: the matching notation should indeed be:



For the sake of readability, we grouped the ties between equal note values. For instance, in the previous example, we grouped the last two linked sixteenth notes into one eighth note. This allows us to display simpler, more idiomatic and more compact solutions. This simplification step is performed on the Lilypond translation of RTs generated by the grammars described, as a post-processing step, by using simple rewriting rules.

4.3 RT Rewrite Rules

In [17, 18] we have proposed systems of rules for rewriting RTs into equivalent RTs. This includes rules such as $2(\_,\_) \to \_$ or $2(\mathsf{n},\_) \to \mathsf{n}$ or $3\big(2(x_1,x_2), 2(x_3,x_4), 2(x_5,x_6)\big) \to 2\big(3(x_1,x_2,x_3), 3(x_4,x_5,x_6)\big)$.
We have considered using RT rewriting for the exploration of equivalent rhythm notations. However, we gave up due to the complexity of this syntactic approach. Rewrite rules as above can indeed be applied at any node of a RT, as long as the subtree at this node matches the left-hand-side of the rule. This gives generally, for a given non trivial RT, many choices of rewrite positions and induces a high divergence in the application of rewrite rules. To be applicable, some restrictive rewrite strategy must be considered and this strategies has to be proven complete.

In comparison, we found the above semantic approach, based on rhythmic values, dramatically less complex.

5. POLYRHYTHMS

We present an application of our approach to obtain good notations for polyrhythms, following studies in [19]. Polyrhythms occurs when two or more rhythmic figures that are not perceived as deriving directly one from the other, or as simple manifestations of the same meter are played simultaneously [20]. In our context, it can be seen as having simultaneously two RTs that do not share the same root-level arities.

Merging two RTs that have different arities is not a trivial task when considering only the trees. However, merging two rhythms is trivial when considering only their onsets: we only have to merge the two onset lists into one list, and order onsets from the smaller to the greater.

To notate polyrhythms, we thus go around the problem of merging trees by going through the time domain. We first convert the two RTs into their rhythmic values, from which we obtain the corresponding onset lists. We then merge the two onset lists into one onset list, from which we get a merged rhythmic value (as an IOI sequence). We then enumerate the equivalent notations of this merged rhythmic value to obtain RTs corresponding to the merged polyrhythm. A schema of this workflow can be found in Figure 5.

The rhythmic value obtained by merging two rhythmic values σ_1 and σ_2 will be denoted as $\sigma_1 \parallel \sigma_2$. Intuitively, $\sigma_1 \parallel \sigma_2$ corresponds to the sequence of durations obtained by playing both σ_1 and σ_2 at the same time.
Let us reformulate the problem of interest :

given a weighted rhythm grammar $\mathcal{G}$ and two non-empty sequences of positive rational numbers σ_1 and σ_2,

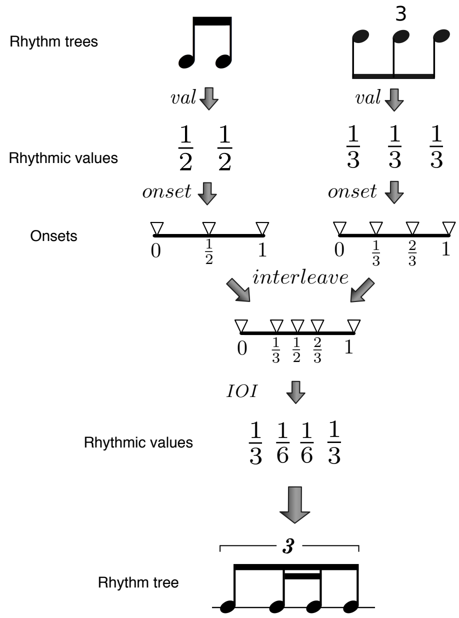


Figure 5. Workflow to obtain notations corresponding to a polyrhythm.

return a weighted rhythm grammar $\mathcal{G}_\sigma$ such that
$\mathcal{L}(\mathcal{G}_\sigma) = \{t \in \mathcal{L}(\mathcal{G}) \mid val(t) = \sigma_1 \parallel \sigma_2\}$.

Therefore, this problem is very similar to the one we addressed in Section 4 and can be solved with the same procedure.

Example 10 *Figure 6 presents a list of rhythms obtained for* $[\frac{1}{2},\frac{1}{2}] \parallel [\frac{1}{3},\frac{1}{3},\frac{1}{3}]$, *using the complex WRG mentionned in Example 9.*

All we need to do now is to define the $\parallel$ operator. To a rhythmic value $\sigma = [d_1,\ldots,d_p]$ we associate a sequence of onsets $onset(\sigma) = [o_1,\ldots,o_{p+1}]$ defined by $o_i = \sum_{j=1}^{i-1} d_j$ for all $0 < i \le p+1$. Note that $o_1 = 0$.

Example 11 *If* $\sigma = [\frac{1}{2},\frac{1}{6},\frac{1}{3}]$, *then* $onset(\sigma) = [0,\frac{1}{2},\frac{2}{3},1]$.

The inverse transformation, called IOI, associates to a sequence of onsets $\ell = [o_1,\ldots,o_{p+1}]$ the rhythmic value $\sigma = [d_1,\ldots,d_p]$ defined by $d_i = o_{i+1} - o_i$ for all $1 \le i \le p$. Finally,

$$\sigma_1 \parallel \sigma_2 = IOI\big(onset(\sigma_1) \cup onset(\sigma_2)\big).$$

Here, the operator $\cup$ denotes the interleaving of sequences of onsets, performed as expected to respect the ordering of onset values.

During the above processing, a special care must be given to the handling of the events associated to the initial rhythmic values σ_1 and σ_2, in order to reassign them properly in $\sigma_1 \parallel \sigma_2$ (see Section 2.3). We leave these details out of this paper.

Example 12 *Figure 7 shows examples of rhythms obtained for* $\sigma_1 \parallel \sigma_2$ *with* $\sigma_1 = [\frac{1}{5},\frac{1}{5},\frac{1}{5},\frac{1}{5},\frac{1}{5}]$ *and* $\sigma_2 = [\frac{1}{3},\frac{1}{3},\frac{1}{3}]$ *corresponding to a split of Violin I part in Stravinsky's Rite of Spring.*

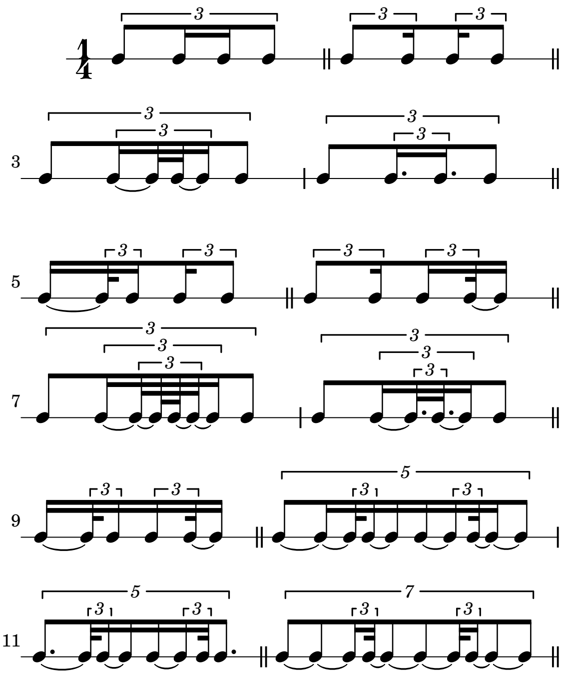


Figure 6. RTs for the merge of $[\frac{1}{2}, \frac{1}{2}]$ and $[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$.

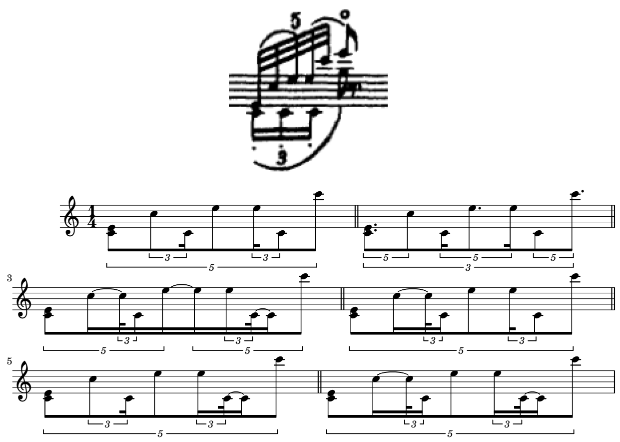



Figure 7. Stravinsky, The Rite of Spring (Violin I) and 6 alternative notations.

Note that in principle, we could merge two rhythmic values that are not the same length. Nevertheless, here, a rhythmic value is always normalized to have a total duration of 1.

6. CONCLUSION

We proposed a formalism based on formal grammars to define languages of weighted rhythm trees, and a procedure to lazily enumerate these trees by ascending weight. It is applied, via grammar constructions, to the problem of enumerating rhythm trees defining the same given sequence of durations (IOIs), and enumerating merges of two polyrhythmic voices into one.

This has been implemented in C++ [1], with command line prototypes outputing the enumerations of RTs which were then translated into a graphical representation in Lilypond. The Lilypond code was moreover post-processed for the improvements described at the end of Section 4.2.

After evaluation with this proof-of-concept prototype, this implementation will be integrated as a dynamic library in OpenMusic, for use in particular as a backend procedure for the transcription framework of [8].

Acknowledgments

The authors would like to thank Karim Haddad for his valuable knowledge and advices on rhythm and notation.

7. REFERENCES

[1] A. Klapuri *et al.*, "Musical meter estimation and music transcription," in *Cambridge Music Processing Colloquium*, 2003, pp. 40–45.

[2] J. Bresson, C. Agon, and G. Assayag, "OpenMusic: visual programming environment for music composition, analysis and research," in *Proc. of the 19th ACM Int. Conf. on Multimedia*. ACM, 2011, pp. 743–746.

[3] A. T. Cemgil, P. Desain, and B. Kappen, "Rhythm quantization for transcription," *Computer Music Journal*, vol. 24, no. 2, pp. 60–76, 2000.

[4] H. C. Longuet-Higgins and C. S. Lee, "The rhythmic interpretation of monophonic music," *Music Perception: An Interdisciplinary Journal*, vol. 1, no. 4, pp. 424–441, 1984.

[5] J. Pressing and P. Lawrence, "Transcribe: A comprehensive autotranscription program," in *International Computer Music Conference Proceedings (ICMC)*, 1993, pp. 343–345.

[6] C. Agon, G. Assayag, J. Fineberg, and C. Rueda, "Kant: a critique of pure quantification," in *International Computer Music Conference Proceedings (ICMC)*, 1994, pp. 52–59.

[7] C. S. Lee, "The rhythmic interpretation of simple musical sequences: Towards a perceptual model,"

[1] http://qparse.gforge.inria.fr

Musical Structure and Cognition, vol. 3, pp. 53–69, 1985.

[8] A. Ycart, F. Jacquemard, J. Bresson, and S. Staworko, "A Supervised Approach for Rhythm Transcription Based on Tree Series Enumeration," in *International Computer Music Conference Proc. (ICMC)*, 2016.

[9] M. Laurson, "Patchwork: A visual programming language and some musical applications," Sibelius Academy, Helsinki, Tech. Rep., 1996.

[10] C. Agon, K. Haddad, and G. Assayag, "Representation and rendering of rhythm structures," in *Proceedings 2d Int. Conf. on Web Delivering of Music*, 2002.

[11] M. Laurson, M. Kuuskankare, and V. Norilo, "An overview of PWGL, a visual programming environment for music," *Computer Music Journal*, vol. 33, no. 1, pp. 19–31, Mar. 2009.

[12] L. Huang, "Advanced dynamic programming in semiring and hypergraph frameworks," in *In COLING Tutorial*, 2008.

[13] H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, C. Löding, D. Lugiez, S. Tison, and M. Tommasi, *Tree Automata Techniques and Applications*. http://tata.gforge.inria.fr, 2007.

[14] I. Shmulevich and D.-J. Povel, "Complexity measures of musical rhythms," *Rhythm perception and production*, pp. 239–244, 2000.

[15] C. Agon, G. Assayag, J. Fineberg, and C. Rueda, "Kant: A critique of pure quantification," in *Proc. of ICMC*, Aarhus, Denmark, 1994, pp. 52–9.

[16] L. Huang and D. Chiang, "Better k-best parsing," in *Proc. of the 9th Int. Workshop on Parsing Technology*. Association for Comp. Linguistics, 2005, pp. 53–64.

[17] P. Donat-Bouillud, F. Jacquemard, and M. Sakai, "Towards an Equational Theory of Rhythm Notation," in *Music Encoding Conference* May 2015.

[18] F. Jacquemard, P. Donat-Bouillud, and J. Bresson, "A Structural Theory of Rhythm Notation based on Tree Representations and Term Rewriting," in *Proc. 5th Int, Conf. on Mathematics and Computation in Music (MCM)*, Springer LNAI vol. 9110, 2015.

[19] K. Haddad, "Fragments de recherche et d'expérimentation: Eléments de réflexions autour de l'écriture rythmique d'Emmanuel Nunes," *seminaire Mamux*, Ircam, 2012.

[20] D. M. Randel, *The Harvard dictionary of music*. Harvard University Press, 2003, vol. 16.

"DES PAS SUR L'INVISIBLE"
THE OCTAVE SPACE AND THE SELF-MULTIPLICATION PROCESS

Sílvia Mendonça
Universidade de Aveiro / INET-MD
silviat@ua.pt

ABSTRACT

The purpose of this paper is to describe the process, from a composition standpoint, from which my piece *Des pas sur le invisible* (2016) for clarinet or solo saxophone, was composed. This work is part of a PhD in music in which I propose a model of composition based on a self-multiplication process, and was created within the context of the Frederico de Freitas Interpretation Prize, Universidade de Aveiro (May 2016 edition).

Starting from a pre-composing point of view, we will consider the octave musical interval as a metaphor for the self-multiplication process. This reflection allows us to think the octave as a space and therefore, how the attribution of this extended dimension can be rethought in music, leading us to new approaches of the composition practice.

The piece *Des pas sur le invisible* will show how this approach can be accomplished, serving to illustrate a thought that takes place outside the proper world of musical elements and considerations that can be decisive in the musical discourse. It will show how the principle behind the conception of this work can develop perspectives for the composition notation practice and for future research.

1. INTRODUCTION

In this reflection we consider the temporal extension of the compositional reality, in order to represent it through a set of practices that rethink some of the compositional resources and parameters, which can also serve the theoretical necessity that manifests itself in the contemporaneity of this practice.

From the composer's standpoint, compositional reality originates from the encounter with the other, and it is projected in a movement from the exterior to the interior. Therefore, there is the possibility of the conception of a gesture to be seen not as the materialization of an idea but, precisely the opposite, as an immersion in the space of conscience, which places the idea itself in the horizon, i.e. the limit.

This piece explores that thought from the concept of unison: a metaphor of the encounter, a starting point to think about self-multiplication, considering the octave as representation of the extended unison. Therefore, a treatment is developed and applied to the musical discourse that consists in filling that space, raising some questions about musical notation.

2. THE OCTAVE: SELF-MULTIPLICATION METAPHOR

It is from a musical gesture that we construct the self-multiplication metaphor, precisely because this self to which we refer is consciousness itself. Consciousness can only be thought from time, and more pronouncedly, from movement. But time cannot be stopped or even fixed, leading us to take the unison from the musical tuning point of view as an ephemeral or even non-existent reality: a paradox. Consequently, the self to which we refer is a reality impossible to fix in the unity of the instant. The self is multiple, existing, from the temporal point of view, in succession.

"The vigil fullness takes place in man when he appears: before others, before himself. So he is with others because he is with himself. He lives in the present that is being present, being sustained in this present that is reiterated in uninterrupted acts: he succeeds himself."[1]

The succession to oneself has a direct correspondence with the multiplication process. It happens to itself, multiplying itself: unit that happens multiplying because it intends to fix itself at every instant.

"Unity is indifferent to production, and generates (engenders) nothing but itself. Unity either divided or multiplied will only create another unity."[2]

In this context, we find a way to speculate about the self-multiplication process, since in order to realize this phenomenon - the multiplication - consciousness needs to unfold, to see itself from the outside. It is therefore essential that a principle of reflection be given to it.

"The spirit, therefore, is the idea that in its return to itself seeks to be fully realized, no longer as a negative of pure exteriority, but taking place in a free world, that is, second nature. The spirit appears concretely in man, for he is the thinking being capable of producing another nature. The spirit, as a man, puts his world "as something reflecting upon itself," takes from nature the character "of another before him," and makes it, rather than something opposite, something made by him."[3]

The correspondence that we find between this second nature of consciousness, according to a reflective principle, and the octave phenomenon, whether it is or not an abstract

[1] ZAMBRANO, M. Pp. 32.
[2] RAMEAU, Jean-Philippe. P. 86.
[3] SOUZA, Roberta Bandeira. Pp. 280-1.

convention, leads us to rethink seriously the very notion of musical interval. The octave is the recognition of the extended unison.

"The proportion of the whole to its half or of the half to the whole is so natural that it is the first to be understood. This should predispose us in favor of the octave, whose ratio is 1:2. The unit is the source of numbers, and 2 is the first number; there is a close resemblance between these two epithets, source and first [Fr. Principe and premier], which is quite appropriate. Likewise, in practice, the octave is characterized by the name "replicate", all replicates being intimately connected to their source (...)."[4]

It is through the central role of the octave, which manifests itself as extension and duration, that we propose to describe the creation of a limited space by polarizing sounds formed on the basis of this interval. In this sense, this is a space that is distributed in height planes representing different cosmos, which are distinguished precisely by the differences of thinking and consequently of representing it, of (re)producing it.

3. THE UNISON AND THE INVISIBLE SPACE OF THE OCTAVE

"Furthermore, the octave serves as a limit for all intervals, so that everything generated by the division of the source, after having been compared to this source, can also be compared to its octave. (...) It is thus manifest that every number multiplied geometrically always represents the same sound, so to speak, or rather gives the replicate of that sound which is its root."[5]

It is because of the impossibility to fix the space between two precisely equal sounds - in unison - that we choose to take the space delimited by the octave as a space that is unfixed, and in an extended sense, unrepresented. In fact, we are actually considering it as a space that exists but, it is not represented, it is invisible. It is a space that has to be covered, but the way to fill it is not defined, and therefore, it is indeterminate, timeless. In order to do that, we will have to fix it somehow and establish references to do so: we have to make sense of it. It is this need for meaning that makes it an interior space: being a space in between, makes it an interior space.

"When the temporality ends, the human being closes itself and opens, thus opening up within him cracks that do not correspond to the different planes in which the vigil unfolds, we would say, in combat formation. The spiral coils on itself and the consciousness then appears in some special point cutting off what is together in the vigil, separating what is united, mixing what is separated according to the images and, what is even more decisive, in function of time itself." [6]

The octave is a sort of shell, a skin, because it is a limit. On one side, the exterior space, on the other, the inner space. Both are considered. We take as reference the various octaves from the same sound (harmonics), that is, their multiplication, for the formation of distinct planes. Each plane represents a new dimension to be explored, precisely from these sounds that will act as polarizing sounds.

"No experience appears alone, disconnected like a lone star. The one that appears on the horizon of consciousness is the center of others that revolve around it or accompany it paling in its light or illuminating in its flicker."[7]

We take the polarization concept as reference. It is the time passage, which allows the fixation and crystallization of events in our memory, so that they can be used in the future. Therefore, the space of the octave is opened, being also a space of freedom, since it unfolds conceptually to infinity for the return of a constructive idea.

4. DES PAS SUR L'INVISIBLE

Originally, the idea of exploring the octave space arises with the desire to write a piece for overtone flute: Fujara. This instrument, without finger holes, and tuned to a fixed frequency, only allows to control harmonics of a fundamental sound through the intensity that is blown into the tube. The main challenge would be to go through these sounds by sweeping the space and by varying the fundamental parameter of intensity.

Afterwards, this same thought process was extrapolated to another composition work for transverse flute. The first challenge here was the definition of the fundamental note, because in this case there is no longer the same rigidity of the overtone flute. The interior of the octave itself can be traveled with greater definition in terms of heights, since it is a chromatic space.

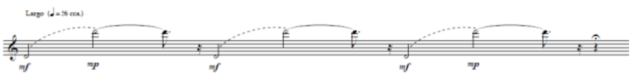



Figure 1. Initial gesture of *Des pas sur l'invisible* for transverse flute.

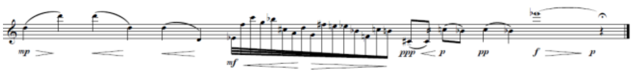



Figure 2. Final gesture of *Des pas sur l'invisible* for transverse flute.

Later, came the opportunity to write for clarinet or saxophone and the possibility of continuing to explore the initial idea, but now with the added challenge of using these transposing instruments. This piece introduces, at the composition level, a new thought over the conception and writing of time and heights, introducing some indeterminate elements, thus giving freedom to the performer in its interpretation.

The three pieces adopt the same name: *Des pas sur l'invisible*. They all have the same starting point idea and take the limit of the octave as the principle of composition, also with the same point of arrival as the main focus: there is always a movement to approach this limit, since the notes that confine it are polarizing notes. But it is above all, a continuous movement because once this limit has been

[4] RAMEAU, Jean-Philippe. P. 82.
[5] Ibid, Pp. 9-10.
[6] ZAMBRANO, M. Pp. 93.
[7] Ibid, Pp. 79.

reached, it is transformed and gives us precisely a new challenge that comes from overcoming it.

We will then, as a result, analyze the work *Des pas sur l'invisible* for clarinet or saxophone. The piece was written to be performed by clarinet or saxophone but also by any instrument of those families. This feature that precedes the conception of the work, allows us to return to the speculations mentioned in the previous chapter, since we have considered that the space of the octave is unregistered and therefore, an indeterminate space.

5. DES PAS SUR L'INVISIBLE FOR CLARINET OR SAXOPHONE

The score comprises of three staves, corresponding to three main polarizing sounds: D4; D5 and D6 (transposed sounds); respectively, the fundamental frequency, the 2nd harmonic and the 4th harmonic. This way of representing the different sounds underlines the importance of the octave interval between them, on the one hand, as the limit of a space to be filled or traveled, and on the other hand, as an intensive space: where different tensions, movements and temporalities are generated:

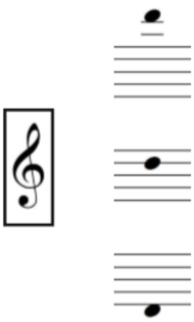


Figure 3. Octave space.

To the space of each octave is initially assigned a particular atmosphere. As we traverse the three planes of the piece, the instrumental timbre changes, as the heights become less defined by the type of explored articulation.

5.1 First Space

The lower octave is the origin, it represents the first plane to be presented, and it is the plane of the fundamental frequency, the deepest plane, where all possible "selves" potentially exist. The musical material consists of a polarizing sound - D4 - and it is chromatically manifested according to the interval module of Bb3 to Ab4 (minor seventh). It is not a complete octave and due to its incompleteness, the fact that it is an incomplete space turns it into a space of unfolding, of transcendence – the will to leave itself. It is characterized by measured vibrattos, trills, air sounds, sung sounds.

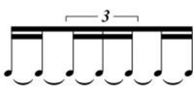


Figure 4. Measured vibrato.

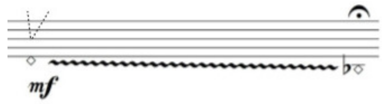


Figure 5. Sung sound.

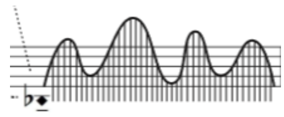


Figure 6. Air sound.

5.2 Second Space

The intermediate plane represents the first octave of the fundamental frequency as a polarizing sound - D5 - corresponds to the plane where the self temporarily affirms, interrupting from time to time the sound of the fundamental frequency, containing sounds that do not correspond exactly to a tuned octave and using quarter tones and tremolos to fill the space. The interval module to be explored is B4 to F5. Because it lies between two planes, a lower one and higher one, it is a compressed zone whose tension is precisely created by the use of microtones and by the range of an augmented fourth as the limit of the musical module.

Figure 7. Quarter tone.

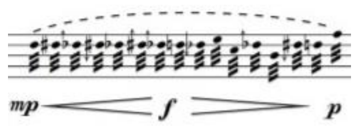


Figure 8. Tremolos.

5.3 Third Space

In the third plane, with the polarization displaced one octave above the anterior plane - D6 - the idea of opening, expansion and transcendence is again manifested, where the musical module now appears from G5 to G6, a complete octave. In this space it is also reached the extreme of the highest register to be explored. However, although this space is the most complete, because it comprises chromatically the interval of the octave, it is an unstable space, of impermanence, of transcendence. This feature of space is explored in terms of timbre by the use of glissandi and staccato.

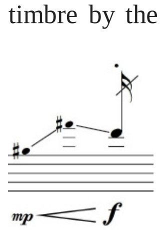


Figure 9. Glissandi and staccato.

5.4 Time, continuity, duration

"Seen from the timelessness of the dream, time is overture, a way of access and way to walk."[8]

The indications concerning the time durations are distributed in several ways. First and foremost, there are blocks of durations indicated below the score. These are general durations, moments. Each moment represents a formal variation that occurs in the discourse. These blocks are a representation of the very structure or shape of the piece, divided into four parts, as shown in Figure 10. It is up to the performer to manage the time of each section according to the characteristics of each sound space.

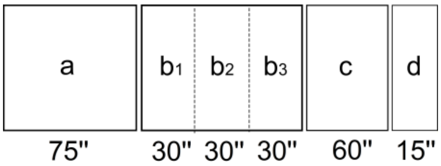


Figure 10. *Des pas sur l'invisible* form.

5.5 Section a

Duration: 75 seconds.
First Space: The same sustained note (fundamental sound) with measured vibrato variations.

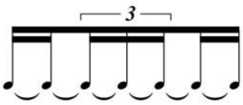


Figure 11. Measured vibrato variations.

Second Space: First space octave with microtonal variations of quarts of tone. They are interruptions of sustained notes but of shorter duration.

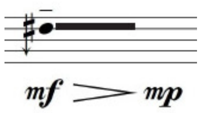



Figure 12. Microtonal variations.

Third Space: Very short attacks on *forte* that intend to leave outside the previously presented plans.

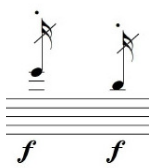


Figure 13. Short attacks.

5.6 Sections b1, b2, b3

Duration: 90 seconds.
First Space: What previously corresponded to the measured vibrato develops in the sense of a greater sound activity with the trill of minor second ascending and groups of demisemiquavers decelerating. It culminates with the arrival of a space of transcendence, lower than the previous polarizing note D4. This note is now Bb3 and is reached through the timbric dissolution of the instrument's sound, consisting now of a sung sound: a glissando from D4 to Bb3.

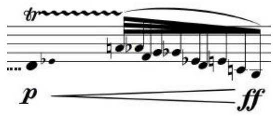


Figure 14. Measured vibrato development.

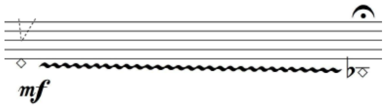


Figure 15. Glissando from D4 to Bb3.

Second Space: We continue to hear the punctual microtones, which in B3 hold for longer, but now on the glissando note sung in first space. A gesture presents a succession of tremolos within the scope of augmented fourth. This zone pretends to be a space of simultaneity and conflict, mixing the two planes.

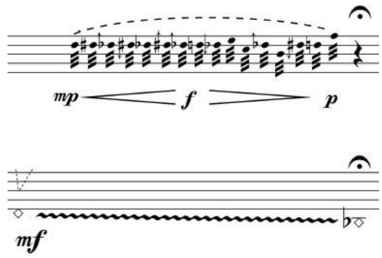


Figure 16. Tremolos succession over a glissando sung.

Third Space: The same short staccato attacks are now articulated with small glissando gestures at increasing intervals between the b three subsections, as an affirmation of the non-permanence of this register. This is a more unstable and also less defined register zone, although these characteristics give it a greater timbric elasticity.

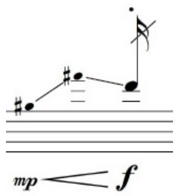


Figure 17. Small glissando gestures.

[8] Ibid, Pp. 76.

5.7 Section c

Duration: 60 seconds.
The growing movement created in each space, precisely because it signifies the register overture, culminates in the emptying of the identity of each plane. The same gesture is graphically explored for each polarizing note. The microtones are no longer heard, and all the D´s are reached with appoggiaturas formed by the notes of the extremes of each interval. The whole speech tends to merge into a single gesture, which ends with the enlargement of the total space reaching the low and high extremes of the register. This originates the arrival of a new fundamental frequency - Bb3 - the motto for section d of the piece.

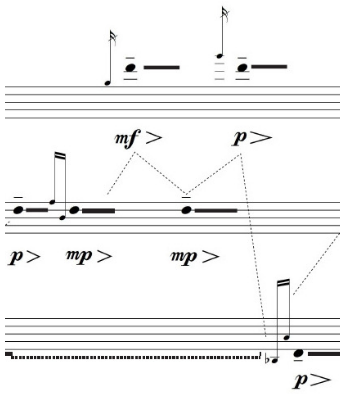


Figure 18. The register overture.

5.8 Section d

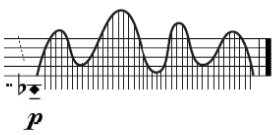


Figure 19. Air sound.

Duration: 15 seconds.
Section d, the smallest, summarizes the whole movement of the piece. It represents the opening of an oscillatory space over a fundamental sound. This sound that emerges stabilized as such in the end of the piece (B3), is also a detuning of the initial sound (D4). As if through movement there was also a descent or fall, a consequence of its own gravity, the non-permanence and ultimately, the ephemerality of the instant.

6. CONCLUSIONS

Although we consider the polarizing sounds, its boundaries and connections by octave intervals, it is when we traverse through this space that we give it an identity, we crystallize it, even if it is done in an imperfect, circumscribed and temporary form. The space of the octave results in an archetrope of the unison idea – metaphor for encounter and instant too.

These moments transform the space into movement or, if you prefer, into blocks of movement - modulation of time - time of consciousness and affirmation of the singular, the instant, the self. The "I" that can only be thought from an interior space - the space of consciousness. Only what is in consciousness actually exists.

The piece *Des pas sur l'invisible* is a way to conceive an indeterminate space, an inner space, a space in between, in constant reformulation by the search to know and to fix its own limits. This boundary lies on the border between outer and inner, which is also the border of the visible and the invisible.

It is up to the subject/spectator/performer of the real to analyze this visible and implied in order to formulate the invisible as it seems/appears to him.

7. REFERENCES

[1] ZAMBRANO, Maria. 1994. Os sonhos e o tempo. Relógio d’Água. Lisboa.

[2] RAMEAU, Jean-Philippe. 1971. Traite de l'harmonie reduites a ses principes naturels. Courier Corporation. https://books.google.pt/books?id=TdU-trQ1cJLgC&n um=13. (24.02.2016, 18h15p.m.).

[3] SOUZA, Roberta Bandeira. 2010. A luta do espírito na natureza: a vitória da liberdade in A noiva do espírito. Natureza em Hegel. Organizadores: UTZ, Konrad Utz e Marly Carvalho Soares. Edipucrs. Porto Alegre. http://ebooks.pucrs.br/edipucrs/naturezaemhegel.pdf (25.05.2016 21:43 a.m.).

VEXATIONS OF EPHEMERALITY

EXTREME SIGHT-READING IN SITUATIVE SCORES - FOR MAKERS, PERFORMERS, AUDIENCES.

Sandeep Bhagwati
Concordia University Montréal
sandeep.bhagwati@concordia.ca

ABSTRACT

What do we do when we subject musicians and audiences to music prompted by real-time scores? Such situative scores create a new kind of immanent relationship between performers and audiences, between composers and performers, composers and audiences – a relationship whose ingrained disregard of context, memory, and knowledge has often been ignored. The use of situative scores seems to inscribe itself into a more general societal trend that uses technology to ephemeralize our lives, to decouple presence from its history. While this immanence has often been perceived as a force for the emancipation of performers and spectators, it can also give rise to unaccountability. Do artistic practices that ephemeralize our artistic 'regime of perception, sensation and interpretation' (Rancière) - such as situative scores – foster abuses of immanence?. In this paper, I will look at such questions from the perspective of the performers, the audiences and the makers of such scores – the composers.

1. INTRODUCTION

Being "in the ephemeral' (Rimbaud) was the dream-come-true of a modernity that erased both the trace and the unconscious, leaving humans without protection or blanket within its transparent walls of glass. [1]

This paper is an exploration of doubts that have plagued me for some time while working on situative scores of various kinds[1][2]– not without a modicum of success over the years, both technological and artistic. Situative scores create a new kind of relationship between performers and audiences, between composers and performers, composers and audiences – a relationship whose specificities, in the rush towards a workable technology, often been ignored, or shelved for future reflection. For me, that future is now. Especially as, in a broader context, the use of situative scores seems to inscribe itself into a more general societal trend that uses technology to ephemeralize our lives, to decouple presence from its history.

While this immanence has often been perceived as a force for the emancipation of performers and spectators, its inherent disregard of context, memory and knowledge can also give rise to irresponsibility, unaccountability and intolerance, especially in the political sphere - as the past year has so amply demonstrated. Do artistic practices that ephemeralize our artistic 'regime of perception, sensation and interpretation' [3] - such as situative scores – mirror or even foster such abuses of immanence?

Situative scores today, especially those relying on digital technology, are structurally oblivious to skill acquisition and training, to transparent perception and analysis, to comparison and re-reading (and, therefore, interpretation), to re-listening and its aesthetic interplay of familiarity and disturbance, to the social aesthetics and taxonomy of sounds, but also to their concrete, emotionally charged materiality. Does this obliviousness tend to abolish the very context that has made these scores arise? Or are such situative scoring practices essentially parasitical[2] [4] in nature - will they always rely on other art practices to provide them with skilled performers and aesthetic contexts of interpretation that they themselves are unable to generate?

In the following, I will look at such questions from the perspective of the performers, the audiences and the makers of such scores. The issues they encounter in contact with situative scores are different in each role. What kind of relationship does 'extreme sight-reading' [5] entertain with the inner dramaturgy and time of the performer? How can an audience understand, evaluate and connect with a performance of a situative score? How does the requirement to meta-compose a situative score, and thus the necessity for a primarily non-linear, conceptual (i.e. not concretely sonic and dramaturgical) approach to composition affect the score maker's musical imaginary? And, as all these roles are intertwined in the

[1] In a previous paper, presented at TENOR 2016 [2], I proposed a taxonomy for situative scores – i.e. scores that, in my definition, do not build on linear, pre-existing and pre-sequenced information. Information in such situative scores is only available ephemerally, i.e. while it is displayed or accessed in a particular context. I proposed four different types of situative scores: 1) rule-based 2) reactional 3) interactive and 4) locative. While types 1) and 4) may be algorithmic in nature, but can also be non-algorithmic, scores of type 2) and 3) are usually not only algorithmic, but also require computer implementation. The subset of situative scores that I am concerned with in this paper are scores of all four types that use algorithms and computer technology to generate and display unforeseeable score information to the musician[s] in real time.

[2] The notion of the 'parasite' here is used as Michel Serres introduces it in his eponymous book [4]: namely that parasites (outcomes that are made possible by an act of communication but belong to neither sender nor receiver) are unavoidable in all communication. He, however, does not see this as unwelcome noise – rather, he reminds us that sometimes the parasites can be much more interesting than the purported focus of the communication. In other words: music academies train musicians for a fixed-score environment; but in the process, they also generate the very players and contexts that make real-time notation at all feasible. Music for situative scores thus is a parasite, feeding on skills which these performers would probably not have developed if situative scores were all that existed. Yet: How long can a performing art sustain itself if it does not educate its own performers?

process of musicking, how do these new demands and affordances, some of them quite categorical, influence each other in the co-emergence (or, possibly, co-effacement) of a new kind of musicking ?

2. EXTREME SIGHTREADING: PERFORMERS

> *"Does everything really have meaning? Aren't there some empty spaces remaining, whose emptiness is perhaps their only meaning? Isn't there a gap there, a hole, between the image produced and the meaning it supplies or dissimulates?"* [6]

Jason Freeman introduced the term "Extreme Sightreading" [5] to characterize the performers' experience with real-time music scores – and to postulate a novel quality of music performance that these scores seem to demand. He discusses a variety of works that seem to highlight four basic generative strategies[3]:

a. ***permutational*** (e.g. Gerhard Winkler's scores where pre-notated elements are re-arranged on the fly, both in time and on the screen),
b. ***parametric*** (e.g. Karlheinz Essl's *Champ d'Action*, where musicians must combine several independently controlled parameters into a comprovisational performance),
c. ***auto-reflexive*** (e.g.Nick Didkovsky's feedback score *Zero Waste* where the pianist's valiant attempt at playing an unplayable score directly generates the next page of this score in an endless open loop)
d. ***co-creative*** (e.g. Jennifer Walshe's or Jason Freeman's works where audience (or conductor) interactions influence how the score appears to the musician)[4]

Most real-time scores, including those I have contributed to or designed (discussed in [2] and [7]), use one or more of these four strategies. While aesthetic concepts and performance strategies may differ, real-time scores all ask performers to subject their musicking to a series of inherently unrehearsable constraints. The comparison most often made here evokes the difference between reading / learning-by-heart the text for a theatre performance and - learning to speak a language.

> When music notation is generated on the fly during live performance, musicians have no opportunity to practice and rehearse the score in advance... As musicians practice a composition, their increasing familiarity with the elements of the notation should help them to perform it more accurately. But that familiarity should also lead them to develop a richer, more personal musical language with which to interpret it. [5, p.34]

Composers and musicians quoted by Freeman share similar metaphors.

> As musicians prepare to perform these kinds of pieces, Gerhard Winkler notes that the process "shifts from 'studying notes' to … [getting to know…'how the system works'" … [Musicians] must not only play the score in front of them as it unfolds, but they must also "bring sense into this succession of un-expectable moments" to create a personal, coherent interpretation of that score. [5],[8]

Or, as a performer describes it:

> "It would have been less interesting if we were totally at the mercy of the notation. But once we got familiar with the process and developed a common approach to the notation . . . then it became more musical" [5, p.35]

What is the interpretation (as Winkler calls it) that a performer can bring to the score? Or, to re-use Freeman's metaphor: once you speak the language of the score - what do you speak about?

'Interpretation' is a term used in the context of fixed scores to describe a process in which practise, repeated readings, analysis, comparisons with other scores, information about the musical or cultural context as well as non-musical concepts and imaginaries are condensed into the moment of performance. Can we really apply this term to situative scores, where neither repeated readings nor comparison are at all possible? While it may be an instance of a larger creative undertaking, each ephemeral performance stands uniquely for itself, is immanent to this particular moment and place. Except for a broad conceptual analysis of the performance context, the performer cannot enter into an interpretational discourse with the score – simply because there is no discursive, coherent or, simply, sequential score outside of the performance.

Whenever I raise such questions in discussions, someone invariably accuses me of underrating the capacity of performers to think on their feet, to analyze what they play as they play. The preferred analogy here are team players in football, rugby or hockey who have barely milliseconds to move their body in a way that will outwit their opponents – in order to perform successfully, they must read the game while it is played. Swordfights are another frequently cited example.

To which there are two types of answers: Firstly, team players can read the game because they know its framework so well, through years of training, that they can perceive and focus on the tiniest variants and aberrations. Most situative score players do not have this luxury, at least not yet. Secondly, in all these endeavours failure is possible (and discernible) - and the failure rate usually is higher than any musician or composer would deem acceptable in the performance of a score. One can either conclude that musicians are better at reading the opponent's (the score's) game than are swordfighters – or, as seems more likely, that they have about the same propensity for failure. Which means that most situative scores are wrongly interpreted most of the time – an appropriate and aesthetically coherent understanding to the elusive score must therefore be a rare and fortuitous event. If one takes this analogy to its logical conclusion, then a musician improvising without any score may have a better chance of making sense of his own performance than a musician performing a situative score.

[3] Please note that the taxonomy offered here is not explicitly stated in Freeman's paper – it is my reading of his text.

[4] Any actual real-time score will obviously combine these four strategies to various degrees and on different levels – I set them out simply as workable analytical categories.

Music performance, by definition, is transient in nature. Sounds disappear, leaving their trace only within our bodies and our memories. Each live performance speaks of the fluidity of existence. In most musicking contexts, however, this ephemerality is counter-balanced by the kinds of immaterial mental architectures, compositions, songs, melodies, rhythms, that become inner entities, part of the software of the mind [9][5]. Usually, these architectures are inscribed in our minds and bodies through constant re-enactment. Repetition and repeatability are salient features of all musicking, and even the most ardent improvisers have their bag of tricks and their somatically and psychologically inscribed, well-rehearsed set of gestures, ideas, concepts.

In this perspective, the extreme sightreading of a continually mutating score implies a double ephemerality: not only must all sound soon die away - the mental architecture of the piece itself, the score and its aesthetic details, specific juxtapositions and inner relationships – all vanish into nothingness as soon as they appear. Is this double ephemerality of real-time score performance a strong artistic acknowledgement of life's general impermanence (as its proponents often claim) – or should it rather be seen as a musical implementation of the built-in obsolescence that underpins most capitalist production and consumption? And is this double ephemerality of performance conceivable as a self-contained aesthetic practice – or must it, structurally and by necessity, sit as a parasite on the simple ephemerality of current musical life? Finally, does this emergent practice demarcate the closing parenthesis of a millenial, eurological score-oriented art music tradition - or does it afford new avenues for critical and aesthetic discourse within this same tradition?

3. TRANSPARENCE & OPACITY: AUDIENCES

> *"Opacities can coexist and converge, weaving fabrics. To understand these truly, one must focus on the texture of the weave and not on the nature of its components. For the time being, perhaps, give up this old obsession with discovering what lies at the bottom of natures… The opaque is not the obscure…it is that which cannot be reduced …"* [10]

Are the above-mentioned critical challenges to the performer of a situative score perhaps compensated by an enhanced or intensified aesthetic listening experience? One could maintain, as indeed makers of situative scores often assert in discussions, that audiences may be afforded new kinds of aesthetic access, as well as new, more emancipated roles in creative musicking - whether they, in following the same score as the players, can aesthetically engage with the difference between score and realisation; or whether they, in actively or unconsciously providing data to the score algorithms, are able to perceive themselves as aesthetic agents within the performance.

> Real-time notation systems, then, offer the opportunity to link the creative activities of listeners to conventional musical ensembles during live performance. This creates a feedback loop in which the audience influences the notation, the notation influences the performers, and the performers, in turn, influence the audience. [5, p.31]

Such co-creative, quasi-iterative loops[6], as well as the ability of the audience to keep comparative tabs on the performers' interpretation, however, introduce a number of novel non-musical factors into the aesthetic experience: like in many games, the interaction itself, its vagaries and rewards, may easily command more attention than the purely auditory experience.

> Some audience members have become so obsessed with the competitive elements emphasized by the animation that the music itself has been relegated to background listening for them." [5, p. 38]

The co-creative feedback loop between audience (or some sort of conductor) and player via sound and score evoked by Freeman, and exemplified in his works Glimmer (2004) and Flock (2007), but also by my own works Native Alien (2009-12) and Fragile Disequilibria (2015) rests on assumptions that invite scrutiny.

Firstly, Freeman himself already notes that a piece like Didkovsky's *Zero Waste* requires an audience of fluent score readers to fulfill its aesthetic goal. Any other listener would simply have no chance of "getting" this piece. Such expert audiences would be Theodor W. Adorno's ideal listeners [11][7]. To all others, the central premise of the piece will remain as opaque as a ritual of a secret sect. But even graphic real-time scores, which prima facie seem to be easier to follow, are not entirely transparent to the audience - not everyone moves between sonic and visual semantics with ease and familiarity. Moreover, traditions may differ in their visual culture as much as in their music.

Regardless of tradition, however, one aspect that characterizes those practices we call art music is their embrace of a sustained and critical discourse as an essential, intertwined strand of their musicking. For such a discourse to be at all feasible, musical utterances must rest on a modicum of convention. As a critic, or a cultured listener, you can only perceive what you already know (and have learned) to be relevant. Be they oral rules or written scores, the quality of a musical rendering within a tradition can only be ascertained by evaluating it against sonic conventions [12][8] that lay down that tradition's specific perceptual, formal and social predilections.

[5] An allusion to Pauline Oliveros's term "Software for People" whose text scores are algorithmic scores open to situative input, but often non-algorithmic in nature.

[6] They are not truly iterative, because the transformation from input to output within at least one of the three stages (the audience) is neither repeatable nor algorithmic.

[7] In his *Introduction to Music Sociology* (1962), Adorno classifies listeners into expert listeners, adept listeners, cultured listeners, emotional listeners and prejudice listeners – the categories describe a decrease in musical expertise in inverse correlation to a rising influence of non-musical factors on the listeners' aesthetic enjoyment.

[8] Shin Yan Sheng calls them "cultural acoustics".

Moreover, in most such traditions, such conventions evolve in parallel with the music, reinforcing any given "style" in a process of autopoiesis - until it seems worth a new generation's while to depart from it, and thus define a new tradition.

Thus, secondly, what are the rules and traditions governing the reception and listening attitudes with regard to real-time scores? Given that this kind of musicking is new even in its comprovisational procedure, which aspect of a real-time score performance would offer inroads for aesthetic criticism and musical engagement? Most reactions that I heard to such a performance focus on the legitimacy of the approach in general. And once that is out of the way, the score-reading strategy, the virtuosity of the musicians interacting with the technology - as if the mere use of a specific technology, or its adequate employ, already conferred aesthetic significance to the resulting sound.

A critical engagement with the sonic content in itself seems rare. I do not remember reading a single *musical* analysis of a real-time score work, maybe for a lack of proper analytic tools. Is this lack of critical engagement with the music itself at all relevant to the practice - or not? Many kinds of music do not need analysis to thrive. Maybe real-time score performance is such a kind of music, upheld by social use, without a layer of critical reflection that would put it into same aesthetic orbit as art music in general? Is it a new apparatus-specific aesthetic sub-genre, similar perhaps to 'orchestra music' or 'electro-acoustic music' or "oil painting" - or does it enter its own, as yet perhaps unnamed and unclaimed, territory of musical styles?

Thirdly, what exactly is the nature of the interaction with the audience in Freeman's ideal feedback loop, where "the audience influences the notation, the notation influences the performers, and the performers, in turn, influence the audience"? Freeman describes a social situation that in itself is not entirely new. Turino [13] mentions village dances in Ghana where the audience 'dances its critique' of the drummers by dancing more or less engagedly, thereby inducing changes in the performance itself. Other traditions, such as Italian opera, khayal, techno, include audience feedback that can serve to guide and, in a limited way, co-create performances.

The innovation brought about by the co-creative score thus seems to reside in the fact that its audience has a more direct access not only to the surface structure of performance, but also to its inner constitutedness, its microstructure - through various interactive schemes and strategies, the audience members may, at least in theory, influence a variety of previously inaccessible musical parameters. But how valid can such a claim to audience co-creativity actually be - given the fact that, as we saw above, the audience, for lack of repeatable and thus interpretable feedback, does not really get the slightest chance to formulate a critical, aesthetically differentiated position vis-à-vis their live-experience.

Even more so in the cases where audiences are <u>not</u> privy to the real-time score, nor get a chance to shape its evolution: the knowledge that the music played by the performer is not the result of a performer's or composer's artistic decisions (whether made in the moment, as in free improvisation, or offline, as in mnemonic or written scores), but of their embodied reactions to a flux of changing circumstance beyond their ken, may significantly shift the import they attach to the aesthetic act of listening. As member of the audience, I sometimes ask myself: where has all the music gone? All I hear is a syntactically vaguely suggestive, sometimes mimetically comprehensible sequence of sounds - but, despite my best and sustained efforts at listening, I cannot engage with them in any critical or even analytic manner. Their very ephemerality seems to belie any message that would go deeper than their performative framing. All to often, I at some point will disengage my critical ear - and simply wallow in the surf of the sound. A different mode of listening, to be sure – but does not the composer's intention, the work of many software engineers and the aesthetic context of this presentation go to waste, if I can only listen to their music as a sonic meditation that maximally offers me opportunities for highly subjective pattern recognition (or, better, pattern invention) ?

The best I can make of some real-time performances is to listen to them as a collateral outcome of an extremely absorbing relationship of the performer with the evolving score, where sonic events are treasured as traces of the body expressive – a perhaps co-creative but, to me, aesthetically opaque loop to which I have very little possibility of access. What do we gain, both in knowledge and in experience, when, instead of lifting the veil of sonic surface that hides musical understanding from us, all we can do is admire the texture of its weave?

4. META^2-COMPOSING: MAKERS OF SCORES

> *The genius is the characteristic product of bourgeois culture…Today, in the period of the collapse of imperialism, any pretensions to artistic genius are a sham.* [14]

Learned eurological composition[9] has largely been a quest for novel exercises in alienation.[15] The perceived need for creators to go beyond their limited selves, to transcend their own contingencies, to questions their instincts and preferences, to escape the strictures of socio-aesthetic conditioning, was a driving force behind the success of notation and many of the conceptual additions to the composer's toolbox that followed it - isorhythm, alpha-numerical coding, *Augenmusik*, serialism, modeling - to name but a few. All of these conceptual strategies abstract the compositional process from purely sonic or aural imagination, transport it to a visual domain, where it can be manipulated and then fed back (via an ever-refined and evolving notation) into the sonic/aural domain. [16] This process of coding and decoding the sonic liberates the composing imagination from sound's intrinsic fickleness and ephemerality, by

[9] I prefer this term to denote what others call "eurogenetic" or, more simply "Western Art" music. "Eurological" encompasses these terms in that it targets all music composition that follows the conceptual logic of eurogenetic music composition – whether it is used by non-Western or even non-human composers, and whether it acknowledges any "genetic" link to Europe or not.

abstracting it from the immanence of the momentary and placing it on a nicely defined operation table. Eurological composition thus usually is a kind of meta-composition[10][17] - an offline intervention into sonic reality.

Its alienation strategies have since been seamlessly extended to provide a strong motivation for artificial intelligence in music and other digital explorations of the sonic. Situative scores, at first glance, seem to be driven by this same impulse: to prepare the elusive sonic for aesthetic consumption by manipulating its conceptual representations. Many situative scores seem to be designed to offer both composers and performers a clinical detachment from the vagaries of actual sound: once more, visual representations are used to describe and denote sonic realities. In fact, all that seems to have changed from the age-old tradition of written composition is a vastly speeded-up process of score generation.

But, of course, in a time-based art such as music, speed is of huge import. With fixed scores, those that offer the most productive resistance to immediate consumption, those that, as it were, slow down digestion, tend to be those that elicit the most sustained engagement. The necessity for practicing, for inscribing a score into the motoric body, becomes a significant factor of meaning production and aesthetic significance. The resistance of a score to both performer and listener is not, as one might surmise, proportional to its undecipherability, nor to the dexterity it demands, but rather to its conceptual complexity, the effort that performer and audiences must make to mentally engage with the multiple meanings afforded by the score: we could call this process "aesthetics-by-resistance".

An emergent score, destined to be ravenously consumed in an act of extreme sight-reading, must by necessity also be a score that offers less resistance (of any kind) to the player. As noted above, it is very likely that the player will skims the score, rather than actually decoding it. He will thus not be able to feel a critical, reflexive *differance* between the score and his sound. As Freeman seems to rejoice:

> "With real-time notation systems, the algorithm and human performer together create a single, merged sonic output." [5, p.36]

This, in turn, means that all the compositional thought that went into creating the ephemeral score will be lost in performance, as the usual 'channel' of musical communication between composer and audience is jammed by the algorithmic aesthetic 'noise' of the situative score. Like the audience that can only admire the texture of the sonic weave without understanding what lies beyond the momentary, composers must resign themselves to being content with meta^2-composition: instead of being cook book writers, they must become cook book - designers. If, as argued above, conventional written composition indead already is a meta-artistic activity, one could label them as meta^2-composers. This embrace of an ever-increasing distance from sonic material sounds uncannily like the beginning gambit of one of those infamous infinite logical regressions, or like the famous ancient political paradox: "Quis custodiet ipse custodiam?"("Who guards the guardian himself?"). Indeed, in situative score performances, the question: "Who composes the real-time score composer?" is both relevant and irrelevant. Relevant, because a score design is indeed always a design decision – and irrelevant, because a better cook book layout does not always lead to the cooking of a tastier meal. And the problem is not that being a designer of cook book layouts, a composer of composers of scores that give rise to music is not an interesting position to be in. It obviously can be - the question is more: whether assuming that role also can be a satisfactory artistic decision. In another article [18], I have indeed argued for the rich artistic terrain that meta^2-composition can afford intrepid composers – and yet: sometimes, in listening to a performance of a situative score that I designed, I feel like an impassioned and successful inventor who went on to found a company based on his ideas - and now spends his all day in administrative and strategic meetings, in activities he would never have wanted to engage in when he started. Do composers of situational scores still have clandestine, torrid affairs with fixed score composition? Alone, at home, do they still tinker around to their heart's content with paper, pen, tablet, softly humming a snatch of music they are just about to write down to keep it from the fate of all things ephemeral – oblivion ?

5. CONCLUSION

"Words about music are like a painted dinner!"
Infamous quip among musicians

A strong sentiment "Against Interpretation" (as in Susan Sontag's eponymous book) [19], complemented by George Steiner's hunger for "Real Presences" [20], has, for the longest time, been a guiding star on my artistic and intellectual path. The joys of unexpected epiphanies, the interest in serendipitous harmonies between seemingly conflicting formal processes, the inexhaustible promises of opacity, the seemingly endless resources of human performers, as well as the speed and diligence of computers still are aspects of an almost childish excitement to be a composer of this century, of my personal now.

Yet recently, in the wake of recent alarming shifts in the political and social atmosphere of the Western world, I began to think about Cardew's contention that avantgarde music serves imperialism [14]. Indeed, the rise in social standing of free improvisation over the 1960s and 1970s has often been associated with the widespread

[10] This term is the same that Robert Rowe and others have used to describe aspects of interactive algorithmic composition. My contention here is that Rowe's "meta-composition" actually is what I would call "meta^2-composition" – the meta-level of a composition practice which in itself is already meta-composing.

unstiffening of western society's spinal columns, and the concomitant, if gradual liberation that has since permeated so many social contracts, always in the direction towards a liberational ideology of ubiquitous individualization and customization of values and social contracts. [21] It is one of the ironies of our time that this inner liberalization requires the exoskeleton of hyper-formal, failure-intolerant systems to 'run'. [22, 23]

Are situative scores not technological incarnations of this ideology, embodying an increasing refusal by sensitive composers to be put on the spot, to be categorized and brought to account? Do they not offer a space of creative indecision for curious performers who mistrust both the know-it-all bullishness of much composed music and the get-it-or-get-out mentality of free improvisation?

If no rules apply, the loudest and strongest prevail. If music cannot be understood in an aesthetical way, other senses will occupy our attention: we will shut down our ears, and we will conceive of everything solely as something to be looked at, for a millisecond – to be instantly forgotten. Instead of all noise becoming music - the dream of the moderns - all music will become noise. It was our wish to make ourselves, and everyone who cares to listen, aware of the beauty, uniqueness and fragility of the ephemeral act. Instead, in an untimely reversal, the ephemeral score, enacting a denial of all musical signification, vexes us with its aggressive absence of meaning, of connection, and of sense: Instead of making our perception more and more aesthetic, its ubiquity of potential aesthetica seems to have created a rich domain for an-aesthetica [24].

I have not yet given up on conceiving a situative score that would allow performers, listeners and composers to collaborate in intellectually and emotionally engaging situative musicking. But I must first find some answers to the many questions raised here.

REFERENCES

[1] Buci-Glucksmann, Christine. *Ésthétique de l'éphémère.* Paris , 2003. p.46 quote translated by Sandeep Bhagwati

[2] Bhagwati, Sandeep, et al. "Musicking the Body Electric. The "body:suit:score" as a polyvalent score interface for situational scores." Edited by Richard Hoadley, Chris Nash and Dominique Fober. *Proceedings of the International Conference on Technologies for Music Notation and Representation - TENOR2016.* Cambridge UK: Anglia Ruskin University, 2016. 121-126.

[3] Rancière, Jacques. *Aisthesis. Scenes from the Aesthetic Regime of Art.* Translated by Zakir Paul. London/NYC: Verso, 2013.

[4] Serres, Michel. *Le Parasite.* Paris: Editions Grasset et Fasquelle, 1980.

[5] Freeman, Jason. "Extreme Sight-Reading, Mediated Expression, and Audience Participation: Real-Time Music Notation in Live Performance." *Computer Music Journal* (MIT Press) 32, no. 3 (2008): 25-41

[6] Zumthor, Paul. *La mesure du monde* . Paris: Seuil, 1993. p. 15. *translated by Burnham/Jenny Coulon*

[7] Bhagwati, Sandeep. "Notational Perspective and Comprovisation." In *Sound & Score. Essays on Sound, Score and Notation*, by Paulo de Assis, William Brooks and Kathleen Coessens, 165-177. Leuwen: Leuwen University Press, 2013.

[8] 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. p.4

[9] Oliveros, Pauline. *Software for People. Collected writings 1963-80.* Baltimore: Smith Publications, 1984.

[10] Glissant, Edouard. *Poetics of Relation.* Translated by Betsy Wing. Ann Arbor: University of Michigan Press, 1997 (orig.1990). p.191

[11] Adorno, Theodor W. *Introduction to the Sociology of Music.* New York: Seabury Press, 1976.

[12] Lee, Yuan-Yuan, and Sin-yan Shen. “The Cultural Acoustics of Chinese Musical Instruments.” In *Chinese Musical Instruments*, by Yuan-Yuan Lee and Sin-yan Shen, 11-15. Chicago: Chinese Music Society of North America, 1999.

[13] Turino, Thomas. *Music as Social Life.* Chicago: University of Chicago Press, 2008.

[14] Cardew, Cornelius. *Stockhausen serves imperialism, and other articles: With commentary and notes.* London: Latimer New Dimensions, 1974. P.53

[15] Bhagwati, Sandeep. "Alienation Strategies in Trans-Compositional Creative Musicking." Edited by Jonathan Impett. *Soundwork. Composition as Critical Technical Practice.* Leuwen: Leuwen UP, in process [2017/18]

[16] Bhagwati, Sandeep. "Notational Perspective and Comprovisation." In *Sound & Score. Essays on Sound, Score and Notation*, by Paulo de Assis, William Brooks and Kathleen Coessens, 165-177. Leuwen: Leuwen University Press, 2013.

[17] Rowe, Robert. *Interactive Music Systems - Machine Listening and Composing.* Cambridge, MA: MIT Press, 1993. See also, by the same author: *Machine Musicianship.* Cambridge: MIT Press, 2001.

[18] Bhagwati, Sandeep. "Musicking beyond Algorithms." In *Patterns of Intuition*, by Gerhard (ed) Nierhaus. New York: Springer, 2014.

[19] Sontag, Susan. *Against Interpretation: And other essays.* New York: MMacmillan, 1966.

[20] Steiner, George. *Real presences: is there anything in what we say?* London: Faber & Faber, 2010

[21] Toop, David. *Into the Maelstrom: Music, Improvisation and the Dream of Freedom. Before 1970.* London: Bloomsbury Academic, 2016.

[22] Small, Christopher. *Musicking. The Meanings of Performing and Listening.* Middletown, CT: Wesleyan University Press, 1998.

[23] Ziarek, Krzysztof. "The Work of Art in the Age of its Electronic Mutability." In *Walter Benjamin and Art*, by Andrew Benjamin, 209-2. London/NY: Continuum, 2005.

[24] Marquard, Odo. *Aesthetica und Anaesthetica. Philosophische Überlegungen.* München: Wilhelm Fink Verlag, 2003

A HIERARCHIC DIFF ALGORITHM FOR COLLABORATIVE MUSIC DOCUMENT EDITING

Christopher Antila
nCoda
christopher@antila.ca

Jeffrey Treviño
California State University, Monterey Bay
jtrevino@csumb.edu

Gabriel Weaver
University of Illinois, Urbana-Champaign
gweaver@illinois.edu

ABSTRACT

We describe an application of hierarchic diff to the collaborative editing of tree-based music representations, using Zhang and Shasha's tree edit distance algorithm as implemented within the XUDiff tool. The edit distance between two trees is the minimum number of edit operations necessary to transform one tree into the other. We consider common operations on the score tree—deleting, changing, and appending tree nodes—to derive a minimal edit sequence, known as an edit script, and we compare the performance of the widely used Longest Common Subsequence algorithm against our approach. We conclude by summarizing implications for the design of collaborative music document software systems.

1. INTRODUCTION

1.1 Collaborative Document Creation Requires Diff Algorithms

In distributed, collaborative document creation, multiple editing agents may change the same original information simultaneously, in complex and overlapping ways. To allow users to resolve conflicting edits and to create a reliable and transparent edit history, robust systems for collaborative editing often depend on *Centralized Version Control Systems* (also known as *Revision Control Systems (RCS)* and *Source Code Management (SCM)* systems). These systems maintain a centralized information representation (*repository*) and a history of users' changes to it; the difference between two edited versions of the information is known as a *diff*, and this difference can be calculated and represented in various ways. Distributed, collaborative music document editing presents unique challenges for the implementation of version control systems, and especially for the implementation of a diff algorithm.

1.2 The Longest Common Subsequence Algorithm as Default Diff

Traditional document comparison algorithms, such as in the Unix `diff` program, take two sequences of characters as input and output an edit script to transform one sequence into the other. An *edit script* consists of a sequence of edit operations (usually insert, delete, and update) to transform the first sequence relative to some entity—usually characters, words, or lines. The *edit distance* is the minimum cost the edit script gives for each operation. The *Longest Common Subsequence algorithm* and its variants are the most common for computing an edit script and cost.

The Longest Common Subsequence algorithm works well in situations where the inputs are sequences of characters and one needs to compare those sequences relative to characters, words, or lines, but many modern file formats rely on hierarchical object models to encode multiple levels of meaning (e.g. XML, blocks of code). As such, different algorithms for hierarchical structures become necessary.

Consider the following two problems that result from this mismatch between character- or line-based diff tools and tree-structured input. First, comparing documents in terms of low-level entities (e.g. lines) may not result in changes that are meaningful to the domain, because lines are often an artifact of presentation: for example, one can generate 'noisy diffs' by just changing whitespace. Second, the manner in which one defines document similarity may change depending upon the task at hand. A poet may get along fine comparing texts in terms of lines, which reflect part of the structure of the text. A musician, however, may want to compare documents in terms of additional information that a line-based approach discards. Our poet, after all, may require a stanza-based representation. Other communities present similarly various demands: scholars may want to analyze and compare texts relative to other structures, such as paragraphs or sections.

1.3 Collaborative Music Information Requires a Hierarchic Diff

These problems are of specific importance to version control for collaborative music document editing, both in terms of usability as well as how one may want to define 'meaning' and 'similarity' in musical information. First, the 'noisy diff' problem—in reporting differences that are not relevant to musicians—creates a usability problem. Although programmers have become accustomed to noisy diffs and the work-arounds they require, the low adoption rate of computer-driven music analytic tools, and the general lack of comfort among music scholars and artists with these tools, suggest that a program producing diffs of meaningless changes would be poorly received by the community. Second, the meaning of textually encoded music always requires additional interpretation, and a one-dimensional sequence of characters (the data structure for which LCS diff was designed) will not allow musicians to compare two different interpretations of the same musical information.

Instead, musicians need the ability to compare musical information in the presence of its logical organization, which must be expressed hierarchically. Therefore, musicians need the ability to compare two versions of a hierarchical structure. Musicians may also want to compare two versions of a score at different levels of abstraction, as represented by these hierarchical structures, or restrict comparison to entities with certain properties: for example, a musician may want to compare two versions of a score in terms of pitch class alone, or of higher-level features like phrase structure. Moreover, a musician may want to filter a score to compare two versions of only a single instrument's staff, or other musical abstractions. For any of this to be possible, diff algorithms must compare edits to tree-based document elements, rather than to document lines or characters.

1.4 Relevant Precedents in the Music and Computer Science Literatures

From the perspective of computer science, our proposed approach leverages previous work from other domains in both industry and in academia. Within industry, there are proprietary, hierarchy-aware difference engines that compare source code in a variety of edit operations, such as the SmartDifferencer by Semantic Designs [1]. Within academia, the tree diffing problem has been long studied by theoretical computer science [2]. Researchers such as Chawathe et al. and Cobena et al. have studied alternative algorithms, such as subtree hashing, and even the use of IDs to align subtrees before similarity computation [3, 4]. We employ the Zhang and Shasha tree difference algorithm to solve the edit distance between trees [5, 6].

While this topic has been approached rigorously in the computer science literature, few relevant precedents exist in the music literature. Almost all software systems for music document creation assume a single user, and collaborative music document creation exists largely as an experimental pursuit. Precedents fall largely into the category of systems for experimental music and interdisciplinary artistic collaboration. Within these experimental systems, version control for distributed workflows has been addressed only implicitly. For example, although Wüst and Jordá track versions in a recursively nested, tree-based structure, in which each successive edit becomes a child node of the version edited, they do not describe algorithms or interfaces for calculating edit distances on this version tree [7]. Likewise, other systems describe distributed collaboration interfaces without addressing the interaction of data representation and version comparison [8, 9].

2. APPLYING THE ZHANG AND SHASHA ALGORITHM

2.1 The Zhang and Shasha Algorithm

Our initial approach uses Zhang and Shasha's tree edit distance algorithm, as implemented within the `XUDiff` tool [10]. The edit distance between two trees is the minimum number of edit operations necessary to transform one tree into another. The edit operations we consider include deleting, changing, and appending tree nodes. As before, a sequence of edit operations between two trees is called an *edit script* and the total number of edits the *edit distance*.

There are several benefits to the Zhang and Shasha tree edit difference algorithm. First, the algorithm produces a tree edit distance that can function flexibly as a metric (assuming the cost function is also a metric). For example, collaborators can use distance metrics to explore the similarity of an entire corpus of musical scores—rather than just two scores—because the metric's notion of distance aligns with our intuition about distance in the physical world. This distance metaphor allows designers to leverage existing musical research in topological feature similarity metrics, which expands the algorithm's utility beyond the notion of hierarchic diff, into new applications such as automated recommendations based on similarity measures [11, 12, 13]. Second, the algorithm is relatively simple and lends itself to a straightforward implementation that can be maintained by an open-source community. The intent of the open-source community is to support an extensible framework for hierarchical comparison of a wide variety of document types across a number of domains. In the future, other algorithms, such as those mentioned above, may be implemented to understand more about the effect of different tree-edit distance algorithms on similarity results.

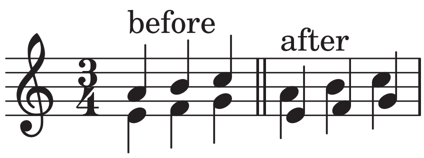


Figure 1. An edit that switches the first and second voices in a staff. Stem direction is the only visual difference, but the underlying representation changes substantially.

2.2 A Comparative Example

Consider the case of a simple edit: exchanging a staff's two voices. That is, as shown in Figure 1, the upward-facing stems of voice one become the downward-facing stems of voice two, and vice versa.

While Common Western Notation displays only a change of stem direction, a tree-based, hierarchic representation of this musical information must alter both the labeling and succession of elements. In the MEI XML representation of a music document, the voice-switch example may be encoded in the following way:

```
<staff n="1">
    <layer n="1">
        <note pname="a"/>
        <note pname="b"/>
        <note pname="c"/>
    </layer>
    <layer n="2">
        <note pname="e"/>
        <note pname="f"/>
        <note pname="g"/>
    </layer>
</staff>
```

After the voice swap, the encoding becomes:

```
<staff n="1">
    <layer n="1">
        <note pname="e"/>
        <note pname="f"/>
        <note pname="g"/>
    </layer>
    <layer n="2">
        <note pname="a"/>
        <note pname="b"/>
        <note pname="c"/>
    </layer>
</staff>
```

2.3 Diff Computation Performance Comparison

This example, although basic, motivates the need to to compare representations of music in terms of hierarchical structure, rather than lines or characters. Figure 2 illustrates an edit script that maps one version of the above MEI-encoded score to another in terms of lines (LCS algorithm). The line-based approach successfully captures the need to exchange the notes between layers; however, the algorithm adds additional noise, because `diff` compares the MEI rather than the hierarchical structure encoded by the MEI. As a result, the total edit distance is 10. If practitioners are interested in understanding change relative to the hierarchical object model of MEI, they will need to sift through the noisy changes produced by a line-based comparison. As mentioned earlier, this may be problematic for widespread adoption within the music community.

In contrast, Figure 3 illustrates an edit script that maps one version of the above MEI-encoded score to another in terms of MEI's hierarchical object model (Zhang and Shasha algorithm). As with the LCS algorithm, the tree-based approach successfully captures the need to exchange the notes between layers; however, unlike the LCS algorithm, the edit distance between individual subtrees has also been summarized. This can be helpful for interpretation, as subtrees closer to the root represent higher-level constructs within MEI, and practitioners can interpret the comparison of the music at multiple levels of abstraction, ranging from low-level notes (six `notes`, each with an edit cost of 1) to higher-level layers (two `layers`, each with an edit cost of 3) and staves (one `staff`, with an

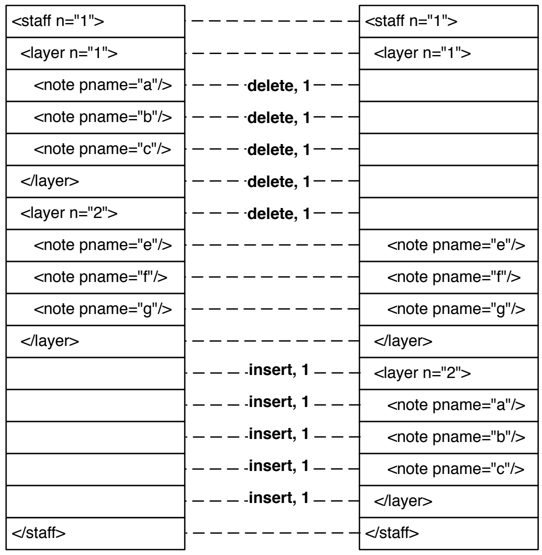


Figure 2. The figure above illustrates the output of `diff` applied to MEI. A total edit distance of 10 results from updating the notes in layer 1 and layer 2 (cost of 6), as well as updating `layer` tags (cost of 4). Nearly half of the edit distance is 'noise' from deleting lines with `layer` tags, an artifact of comparing versions in terms of lines instead of MEI elements.

edit cost of 6). Most notably, the tree-based element-by-element comparison reduces the edit distance to almost half of that of the LCS algorithm: the edit distance has been reduced to 6, from LCS's 10, most of which was noise from deleting lines with 'layer' tags (an artifact of line-based comparison).

3. CONCLUSIONS

The recently emerged potentials of online, collaborative music applications illustrate several ways that a robust, hierarchic diff algorithm for music can enable newly collaborative musicology, composition, and music education through document utilities [14, 15, 16, 17]. Yet the commercial presentation of widely used digital engraving tools still conflates the act of sharing with the act of collaboration, although these remain distinct from each other. As a recent advertisement for the Sibelius engraving program exhorts, 'Collaborate more easily with others and distribute your compositions for the world to hear. Share scores through email, upload and publish them as sheet music on ScoreExchange.com, and even share your composition as a video or audio file on YouTube, Facebook, and SoundCloud' [18]. While file exchange between music authors remains crucial for musical creativity and collaboration beyond notation, it is time for engraving software to embrace the potentials of genuinely collaborative music document editing interfaces. But distributed music document collaboration requires robust, intuitive version control algorithms and interfaces, and designers must reassess the task of music representation in light of the need for hierarchic diff. The superior performance of the Zhang

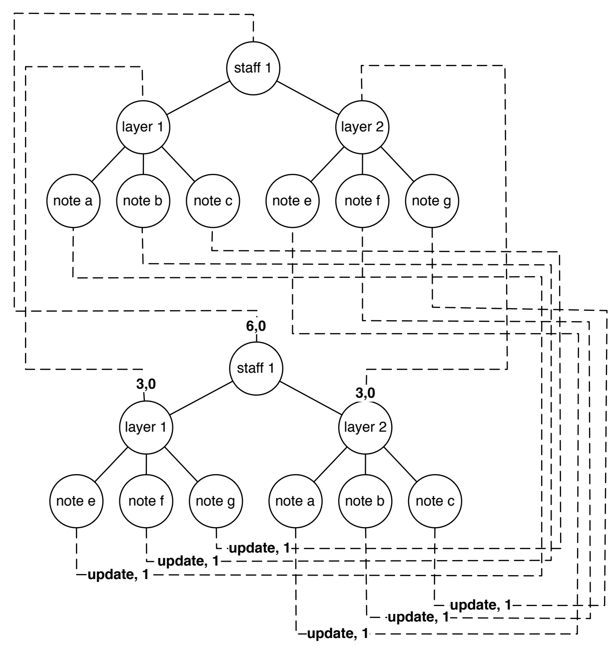


Figure 3. The figure above illustrates the output of `xudiff` applied to MEI. A total edit distance of 6 results from updating the notes in layer 1 and layer 2. Total costs are aggregated across the hierarchical structure of the MEI text.

and Shasha algorithm shown here suggests that purely tree-based representations, such as MEI, should be adopted for collaborative music software systems.

Acknowledgments

Research conducted for the *nCoda* project has been supported by Colorado College's SEGway faculty support grant.

4. REFERENCES

[1] S. Designs, "Semantic Designs: Smart Differencer Tool." [Online]. Available: http://www.semdesigns.com/Products/SmartDifferencer/

[2] P. Bille, "A Survey on Tree Edit Distance and Related Problems," in *Theoretical Computer Science*, vol. 337, June 2005, p. unknown.

[3] S. Chawathe *et al.*, "Change Detection in Hierarchically Structured Information," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '96)*, June 1996, pp. 493–504.

[4] G. Cobéna *et al.*, "Detecting Changes in XML Documents," in *Proceedings of the 18th International Conference on Data Engineering*. IEEE, February and March 2002, pp. 41–52.

[5] K. Zhang and D. Shasha, "Simple Fast Algorithms for the Editing Distance between Trees and Related Problems," *Siam Journal of Computing*, vol. 18, pp. 1245–1262, December 1989.

[6] K. Zhang, "The Editing Distance between Trees: Algorithms and Applications," Ph.D. dissertation, New York University (NYU), 1989.

[7] O. Wüst and S. Jordà, "Architectural Overview of a System for Collaborative Music Composition over the Web," in *Proceedings of the 2001 International Computer Music Conference*. Citeseer, 2001, pp. 298–301.

[8] S. Balachandran and L. Wyse, "Computer-mediated Visual Communication in Live Musical Performance: What's the Score?" in *Arts and Technology*, A. L. Brooks, Ed. Springer, 2012, vol. 101, pp. 54–62. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-33329-3_7

[9] D. Hepting and D. Gerhard, "Collaborative Computer-aided Parameter Exploration for Music and Animation," *Computer Music Modeling and Retrieval*, pp. 158–172, 2005. [Online]. Available: http://www.springerlink.com/index/XJXLV487NLUU9W9V.pdf

[10] G. Weaver, "Security-Policy Analysis with eXtended Unix Tools," Ph.D. dissertation, Dartmouth College, 2013.

[11] J. P. Bello, "Measuring Structural Similarity in Music," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2013–2025, 2011.

[12] A. Berenzweig, B. Logan, D. P. Ellis, and B. Whitman, "A Large-scale Evaluation of Acoustic and Subjective Music-similarity Measures," *Computer Music Journal*, vol. 28, no. 2, pp. 63–76, 2004.

[13] Y. Panagakis and C. Kotropoulos, "Music Genre Classification via Topology Preserving Non-negative Tensor Factorization and Sparse Representations," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2010, pp. 249–252.

[14] D. Martin *et al.*, "LeadsheetJS: A Javascript Library for Online Lead Sheet Editing," in *Proceedings of The First International Conference on Technologies for Music Notation and Representation*, 2015.

[15] P. McCulloch, "THEMA: A Music Notation Software Package with Integrated and Automatic Data Collection," in *Proceedings of The First International Conference on Technologies for Music Notation and Representation*, 2015.

[16] Flat, "flat.io." [Online]. Available: flat.io

[17] T. Bača, J. Oberholtzer, J. Treviño, and V. Adán, "Abjad: An Open-source Software System for Formalized Score Control," in *Proceedings of The First International Conference on Technologies for Music Notation and Representation*, 2015.

[18] Avid, "Sibelius: Features." [Online]. Available: http://www.avid.com/sibelius/features

Establishing connectivity between the existing networked music notation packages *Quintet.net*, *Decibel ScorePlayer* and *MaxScore*

Stuart James, Cat Hope, Lindsay Vickery, Aaron Wyatt
Western Australian Academy of Performing Arts, Edith Cowan University; Sir Zelman Cowen School of Music, Monash University
{s.james,l.vickery}@ecu.edu.au, cat.hope@monash.edu, music@psi-borg.org

Ben Carey, Xiao Fu, Georg Hajdu
Center for Microtonal Music and Multimedia (ZM4)
Hamburg University of Music and Theater (HfMT)
{ben.carey,xiao.fu,georg.hajdu}@hfmt-hamburg.de

ABSTRACT

In this paper we outline a collaboration where live internet-based and local collaboration between research groups/musicians from *Decibel New Music Ensemble*[1] (Perth, Australia) and *ZM4*[2] (Hamburg, Germany), was facilitated by novel innovations in customised software solutions employed by both groups. The exchange was funded by the *Deutscher Akademischer Austauschdienst*[3] and *Universities Australia*[4]. Both groups were previously engaged in the research and performance of similar musical repertoire such as John Cage's 'Five' (1988) and 'Variations I-VIII' (1958-67) among others, the performances of which utilise graphic, animated and extended traditional Western music notation. Preliminary steps were taken to achieve communication between the three existing network music notation packages, the *Decibel ScorePlayer*, *MaxScore* and *Quintet.net,* facilitating a merging – and ultimately an extension – of notational approaches previously prescribed by each music notation package. In addition to the technical innovations required to achieve such a project, we consider the outcomes and future directions of the project, as well as their relevance for the wider contemporary music community.

[1] Retrieved 26th Nov 2016 from http://www.decibelnewmusic.com/
[2] Retrieved 26th Nov 2016 from http://www.hfmt-hamburg.de/forschung/zm4/
[3] Retrieved 26th Nov 2016 from https://www.daad.de/en/
[4] Retrieved 26th Nov 2016 from https://www.universitiesaustralia.edu.au/

1. INTRODUCTION

Although many existing software solutions allow the display of music notation outside the conventions of traditional Western music notation – graphic, animated, extended traditional Western, and real-time notation (refer to Table 1) – a standardised format for communication between these software packages in real-time is yet to be established. Open Sound Control[5] and MIDI have long offered increasingly comprehensive solutions for dealing with sound generation, but have a few shortcomings when it comes to dealing with the transmission of rich graphics required for real-time eScore delivery in live performance [1]. This leaves performers of such music with the choice of only performing music using the prescribed methods employed by existing packages, or innovating their own software solutions. Performers seeking to collaborate with other innovators in the same field face further limitations in that these software solutions are likely to be incompatible due to their software design, communication protocol, and the command syntax adopted.

In order to overcome this problem, the strengths and limitations of the various software packages used by the Decibel New Music Ensemble and the ZM4 research group (respectively the *Decibel ScorePlayer* [2] App[6] for the Apple iPad, and *Quintet.net*[7] [3] and *MaxScore*[8] [4]

[5] Commonly known as OSC. Refer to Wright, M., Freed, A., & Momeni, A. (2003). Open Sound Control: State of the Art 2003.' International Conference on *New Interfaces for Musical Expression*, Montreal, p. 153-159.
[6] Retrieved 26th Nov 2016 from https://itunes.apple.com/au/app/decibel-scoreplayer/id622591851?mt=8
[7] Retrieved 26th Nov 2016 from https://quintetnet.hfmt-hamburg.de/wiki/pages/w7u7v9j3/Download.html

with its recent extension, the *NetCanvas* [5], for computers running MacOS or Windows) were documented (see Table 1) alongside other real-time notation packages to reach a viable standard for inter-application communication, in the fashion of previous efforts such as the *Max-Unity3D Interoperability Toolkit* [6], which utilises TCP/IP socket connections to transfer messages in real-time between *Max* and *Unity3D*, and *PWGL* and *Noteability* which instead support the OSC standard [7].

Some of the primary questions raised in such a collaboration are largely driven by how it may be possible to overcome some of the intrinsic differences of music notation packages in such a way that it does not mean re-inventing the wheel. There are a number of facets of this consolidation that are worth considering: the type of score and the scope[10] of the notation used, whether it is fixed or animated, the nature of networked performance and whether it is intended for both local area network (LAN) and wider area network (WAN) performance situa-

Software	Connectivity	Protocol and Data Type	Traditional Notation	Graphic Notation
Quintet.net (1999)	• Quintet.net Client • Quintet.net Server • Quintet.net Conductor • Quintet.net Viewer	Max messaging via UDP and TCP socket connections	Both Generative and Fixed via MaxScore	Yes. Fixed Animated, Live-Animated, and Live-Generative via MaxScore (refer to Figure 1) [8]
Bach [9] (2012)	No native support	Messaging is possible using Max over UDP or TCP	Both Generative and Fixed	via slot messages, requiring an external drawing object
InScore [10] (2012)	No client / Server architecture per se, rather the InScore application responds to remote OSC messages received	OSC Packets over UDP	Both Generative and Fixed	Yes. Fixed Animated, Live-Permutated, Live-Animated, and Live-Generative
Decibel ScorePlayer (2013)	Decibel ScorePlayer Client / Server over a TCP socket connection	OSC Packets over UDP	Limited	Yes. Fixed Animated, Live-Permutated, and Live-Animated
MaxScore.NetCanvas (Apr-Nov 2016)	• Internet browser application • NetCanvas Server • NetCanvas Maxpatch abstraction • NetCanvas virtual-reality / browser client	Rendered PNGs are streamed as formatted packets of base-64[9] byte arrays over a TCP websocket connection	Both Generative and Fixed	Yes. Live-Animated

Table 1. Comparison of connectivity, protocol, and notation types for existing ScreenScore applications.

[8] Retrieved 26th Nov 2016 from http://www.computermusicnotation.com/

[9] 64-bit

[10] Arguably, in the case of some animated scores, constraints may be necessary, and such solutions may not address the scope of what is possible with openGL and VR notation. The game industry, for example, has already had to define a framework and syntax for animation.

tions, and the technical aspects of integrating both the networking protocol and syntax to allow such software solutions to talk with one other and remain synchronised. These technical aspects follow some key central ideas: the interoperability of digital score solutions, multiple devices, and software platforms; the need to evaluate existing software dependencies of digital score solutions (see Table 2); the need to further extend the feature set of such solutions; and the need to consolidate some of the intrinsic differences between existing software solutions. Such aims of interoperability investigated include:

- Integration and synchronisation of Quintet.net and the Decibel ScorePlayer
- Development of a Canvas score module for the Decibel ScorePlayer and a Max abstraction responsible for converting the draw commands from MaxScore to the Canvas score in the ScorePlayer
- Transmission of OSC via network sockets (over TCP/IP or UDP) to the MaxScore.NetCanvas Server from the Decibel ScorePlayer and Max 7
- Integration of support for OSCblob message packets over UDP to the ScorePlayer app from Max 7, decoded as PNG textures in the Decibel ScorePlayer
- Integration of support for layer creation/deletion, cursor automation, via existing Decibel message syntax in MaxScore.NetCanvas from Max 7

Music Notation Package	Software Dependencies
Quintet.net	Max and various 3rd party externals packages
MaxScore	Max, Java Music Specification Language (JMSL), and 3rd party externals packages
Bach	Max
InScore	An external application capable of sending OSC messages
Decibel ScorePlayer	None
MaxScore.NetCanvas	Max, Java Music Specification Language (JMSL), and 3rd party externals packages

Table 2. Comparison of Music notation Packages and their software dependencies.

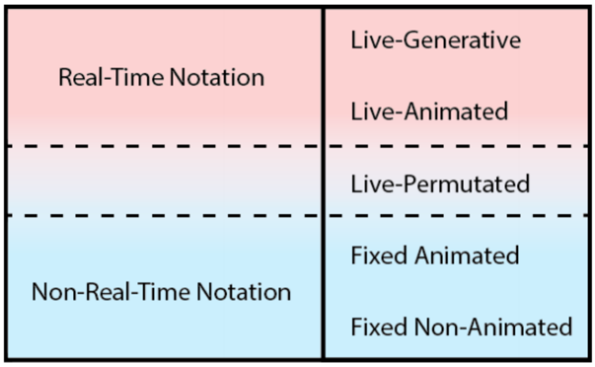


Figure 1. Categories of real-time and non-real-time music notation. (Shafer, 2016).

2. DEVELOPMENTS IN THE DECIBEL SCOREPLAYER

When the Decibel ScorePlayer was first developed, it was conceived of as a standalone application without intrinsic software dependencies. The application specifically focussed on graphic and animated notation on a portable device (the iPad), with the intention that multiple devices could be synchronized in real-time over a Wi-Fi network. The Decibel ScorePlayer is hard-coded in objective-C and the binary application currently supports the iPad generation 2 and 3. The motivations in using such an application were driven largely toward ease of use for the end-user, both composer and performer.

The ScorePlayer adopted the zero configuration networking DNS service discovery [11] method in order to streamline the process of connecting multiple devices over the network. Each device broadcasts a service name determined by the score that is open, and the ScorePlayer application adopts its own client-server method involving the management of both a primary and secondary server and remaining clients on the network. The secondary server is used as a backup primary server in case the existing primary server drops off the network. This networking architecture has proven to be robust in live performance settings. Over a period of some years of using this in live performance, Decibel New Music ensemble have found this to be a robust and reliable solution for synchronously (or asynchronously) presenting scores over multiple devices.

A number of enhancements to the Decibel ScorePlayer were completed as part of the *Deutscher Akademischer Austauschdienst* and

Universities Australia exchange. These included providing WAN capability for the ScorePlayer application, and the completion of several modules for the ScorePlayer allowing the application to function in a number of different modes: a 2D scrolling “Talking Board” mode, a generative notation “Rodinia” Conductor/Performer/Audience mode, and a blank externally controllable “Canvas” mode.

2.1 WAN Internet Connectivity

The ScorePlayer was originally intended for synchronising multiple iPads over a Local Area Network (LAN) [12]. However, one of the first proposed outcomes of the research exchange was to enable the ScorePlayer to synchronise over a Wide Area Network (WAN) for telematic performance. Decibel ensemble member Aaron Wyatt extended the ScorePlayer to facilitate this process, allowing the user to manually enter a destination IP address so that a local iPad, assuming the role of a client via TCP, can remotely connect to another iPad which in turn assumes the role of a server (refer to Figure 3). This development, after some testing, was incorporated as part of a telematic concert on occasion of the Sound and Music Computing Summer School 2016 between the *Hochschule für Musik und Theater* in Hamburg, Germany, *Edith Cowan University* in Perth, Western Australia, and *Stanford University* in California, United States. The concert successfully presented a series of new works by participants of the Summer School featuring performers from three different countries around the world (**Figure 2**).

Figure 2. Three-way connectivity in a telematic performance employing JackTrip for audio streaming and the Decibel Score Player for synchronized score rendering across 15 time zones.

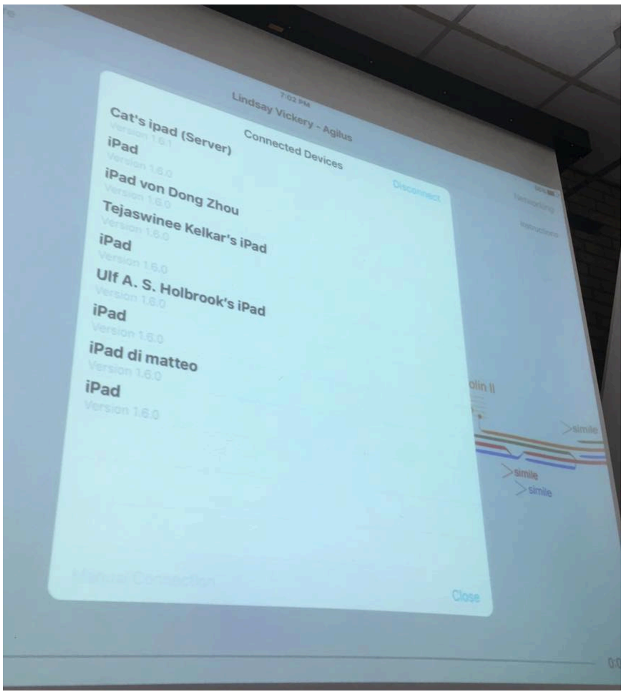


Figure 3. The iPads connected during the telematic performance at SMC2016 in Hamburg.

2.2 Talking Board Mode

In Vickery and Hope’s work ‘The Talking Board’ (2011) [12] a graphical score-collage is continuously repositioned during the performance, moving smoothly in the vertical and horizontal dimensions, and also jumping to particular new positions. The four performers realize the work by interpreting the components of the score that are framed by four colour-coded planchettes (circles). Rather than following defined pathways, the planchettes move in 2-dimensions according to a set of behaviours (wander, converge, flock, lead). An OSC networking protocol was implemented in the ScorePlayer to synchronise the iPad score with an external Max patch [13]. In the “talking board” mode, external communication was introduced to allow data on the score and planchette positions to be used to control spatialisation and processing within Max.

The behaviours are transmitted via OSC using the following messages:

/External/MoveBackground x1 y1
(the xy position of the background collage).
/External/MovePlanchette x1 y1
(the xy positions of each planchette).

2.3 Rodinia Mode

Rodinia is a varied paradigm in the ScorePlayer: a networked, generative and interactive, conducting environment [14] allowing for control by four "conductors" of generative notation for four ensembles. The score includes three view paradigms: audience, conductor and performer. The work draws in part from the generative functions developed for the *Decibel Cage Variations App* (Variations I and II).

Vickery's work 'TECTONIC: Rodinia' (2016) employs a collision avoidant algorithm which may modify the choices of each conductor. As notational streams approach one another they are pushed upward or downward according to their evaluated mass. The controller interface is operated by two hands (the iPad permits 11 simultaneous multi-touch points) [15] allowing parameters to be specified simultaneously by the Left hand (play/hold, articulation, duration type) and Right hand (duration, pitch, dynamic, rate and compass) (see Figure 4).

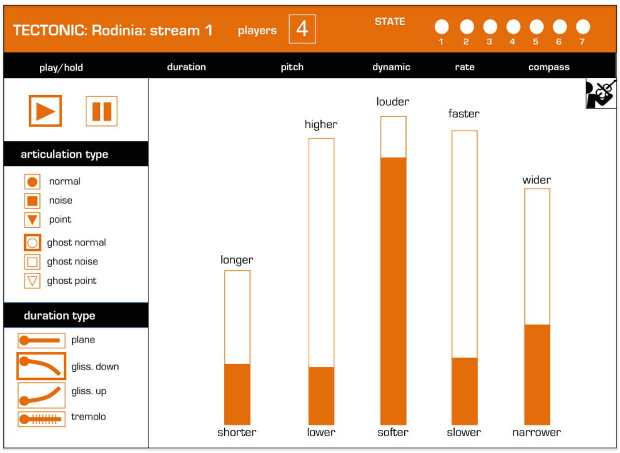


Figure 4. "Conductor View" in Lindsay Vickery's 'TECTONIC: Rodinia' (2016).

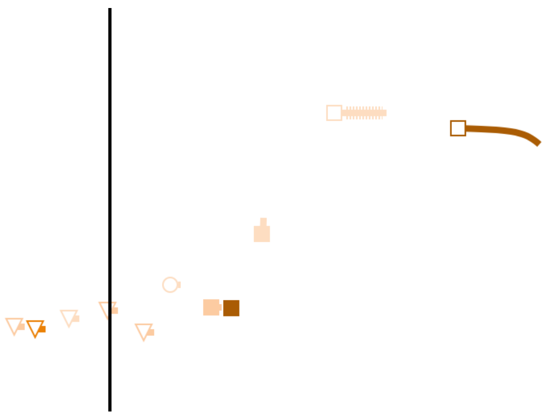


Figure 5. "Performer View" in Lindsay Vickery's 'TECTONIC: Rodinia' (2016).

These parameters define the boundaries of stochastically generated graphical events which are distributed to the all of the iPads belonging to the same stream on the network. Musicians read the generative score in a "performer view" where notation for each of the four ensembles is scrolled right to left across the iPad screen (see Figure 5).

The "audience view" amalgamates the notation from each stream into a single score, to be shown on a large screen behind the performers for both audience and the conductors. Audience view draws the streams of notation approaching from four directions (left, right, top and bottom) (see Figure 6). The notation "wraps" around each time it completes the crossing from one side of the score to the other.

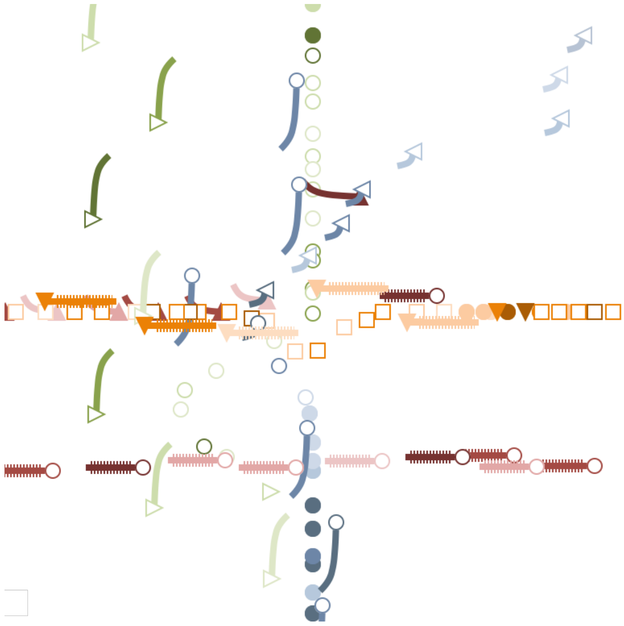


Figure 6. "Audience View" in Lindsay Vickery's 'TECTONIC: Rodinia' (2016).

The full range of conductor defined commands as well as the current xy positions of each part in the "audience view" are transmitted to Max as OSC commands:

/External/Hold – (allowing the generative of notation for a group to be halted).
/External/Articulation – (defining 6 articulation types)
/External/DurationType - defining 4 duration types)
/External/BarValue - (defining duration, pitch, dynamic, rate and compass)
/External/Event – (xy position of each group)
/External/PlayerNumbers (defining the number of players in each group)

This data is used to process and spatialise the sound of each instrumental group according to the conductor defined parameters in real-time.

2.4 Canvas Mode

The Canvas mode principally varies from previous modes in that it accepts external messages defining core elements of a digital score, such as the dynamic creation of layers and loading and positioning of image files. In Canvas mode the ordering, occurrence and motion of the score segments can be determined externally by OSC messages sent via UDP. Previously all score generation – as found in the iPad implementations of Cage ‘Variations I’ (1958) and ‘Variations II’ (1961) or Vickery’s ‘TECTONIC: Rodinia’ (2016) – and indeterminate elements – as found in Hope’s ‘Liminum’ (2012) or Vickery and Rose’s ‘Ubahn c1985: the Rosenburg Variations’ (2012) – were generated using randomised procedures within the ScorePlayer application.

Canvas mode allows composers to implement generative and indeterminate works in the ScorePlayer without the generative procedures having to be hard-coded into the ScorePlayer application. The concept was trialled with a pre-existing score: Samuel Dunscombe's ‘Westpark’ (2012) for bass flute, bass clarinet and electronics.[11] In the original version of the work, the score comprised 46 indeterminately presented images and 4 dynamic markings that were screened on networked laptops. Dunscombe intends the performers to react almost instantaneously to the often rapidly changing images.

The Canvas mode was used to define a background score layout and to place the score images and dynamics within separated parts for each instrument.

A Canvas mode score is defined by an .xml file formatted as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE opus SYSTEM "opus.dtd">
<opus>
  <score>
  <name>Title</name>
  <composer>Composer Name</composer>
  <type>Canvas</type>
  <duration>0</duration>
  <instructions>workinstructions.png</instructions>
    </score>
</opus>
```

This xml file and the required image files are compressed in a .zip file, which is then given the extension .dsz and loaded into the Decibel ScorePlayer via iTunes[12]. Messages to define the score changes from Max are transmitted as OSC packets to the ScorePlayer over UDP. In the Decibel ScorePlayer, a single image must be loaded into a discrete graphics layer. The canvas mode score accepts messages in the following formats:

/Renderer/Create “name_of_layer” n x1 y1 x2 y2
(Where n defines 0 a layer that appears in all parts or 1…n the part in which this layer appears. x1 and y1 define the image (top left) x and y position and x2 and y2 define the image (bottom right) x and y position in pixels.)

/Renderer/LoadImage “name_of_layer”
imagename.png n
(where n is either 1 to auto size the image/0 to use the image’s “actual size”)

/Renderer/Remove “name_of_layer”

/Renderer/SetPosition “name_of_layer” x y
(Where x and y define the top left coordinate of the image in pixels).

Tests of the Canvas mode version (see Figure 7) of the score suggest that the iPad score load images more quickly than the original Max version even though the commands are being transmitted across a network.

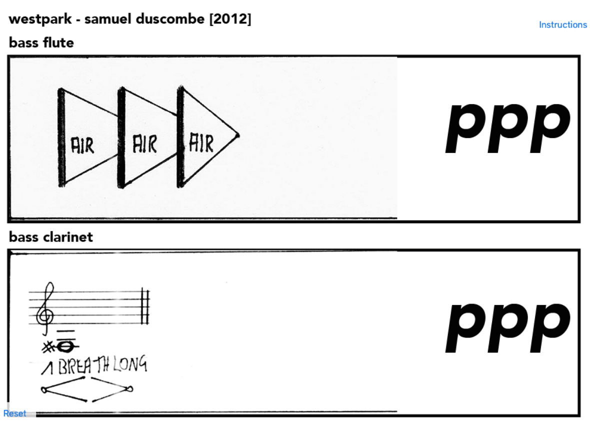


Figure 7. Decibel ScorePlayer version of Samuel Dunscombe’s ‘Westpark’ (2012)

[11] It also was used to create a new version of Vickery’s work ‘abstract clouds of the western skies’ *(2016)* which had been originally authored in MaxScore as part of the exchange.

[12] Retrieved 26th Nov 2016 from https://vimeo.com/140709500

The Canvas mode interestingly permits a greater degree of "animation" than is normally found in the Decibel ScorePlayer as layers can be repositioned and resized independently. This possibility has not yet been exploited in a creative work.

2.5 Interoperability between MaxScore and the Decibel ScorePlayer

One of the further developments in progress as a result of this research exchange has been the investigation as to the viability for MaxScore to use the Decibel ScorePlayer as a canvas window to display generative notation. This process has required sifting through the draw commands Nick Didkovsky has formalised in [mxj com.algomusic.max.MaxScore]. Currently, Georg Hajdu has implemented the MaxScore canvas in Max using JavaScript mgraphics commands. However, as the draw commands in the Decibel ScorePlayer Canvas module are different, an abstraction in Max is necessary and responsible for translating the commands from MaxScore into the appropriate syntax. Furthermore, the display ranges are normalised according to the size of the display window, and some adjustment is made depending on the anchor position of each graphical element displayed on-screen.

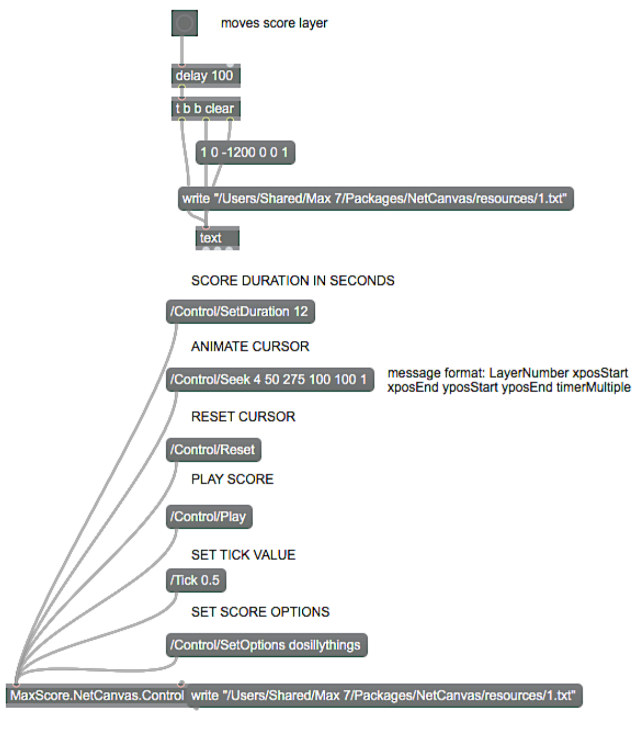
MaxScore	Decibel ScorePlayer
staffLine 0. 0. 0. 0.5 20. 51. 781.79364 51.	/Renderer/Create staffline1 0 24 70 949 60 /Renderer/LoadImage staffline1 StaffLine.png 0
tr 22. 75.96 0.5 Staff 0. 0.	/Renderer/Create trebleclef 0 26 161 176 311 /Renderer/LoadImage trebleclef TrebleClef.png 0

Table 3. Comparison of draw commands in MaxScore and the Decibel ScorePlayer.

The research exchange has also been concerned with the real-time transmission of graphical data over the network. Certain restrictions in the size of an OSC full-packet support the idea of splitting large files into smaller packets. Research is in progress to determine the best way to transmit media over the network.

3. DEVELOPMENTS IN MAXSCORE.NETCANVAS

As a result of this collaboration, and in line with development goals set out for the MaxScore.NetCanvas project, a number of changes were made in anticipation of the release of NetCanvas beta 0.2. Firstly, NetCanvas now supports the creation of multiple graphical layers, which can be animated via command sets that can be created in Max and sent via the NetCanvas Server. Parameters can be changed in real-time via messages that follow the ScorePlayer message syntax (see Appendix 1). This major change allows for scrolling scores, opacity between layers and more exotic animations of layers, which can be updated via websocket connections made in the browser. Secondly, cursor support was implemented following the logic of Georg Hajdu's composition 'Carnage', originally composed for a joint concert between the two research teams in Perth in July 2016. Cursors can be controlled independently, instantiated and destroyed, set to animate in a variety of states and set to different shapes, sizes and colours (Figure 8).



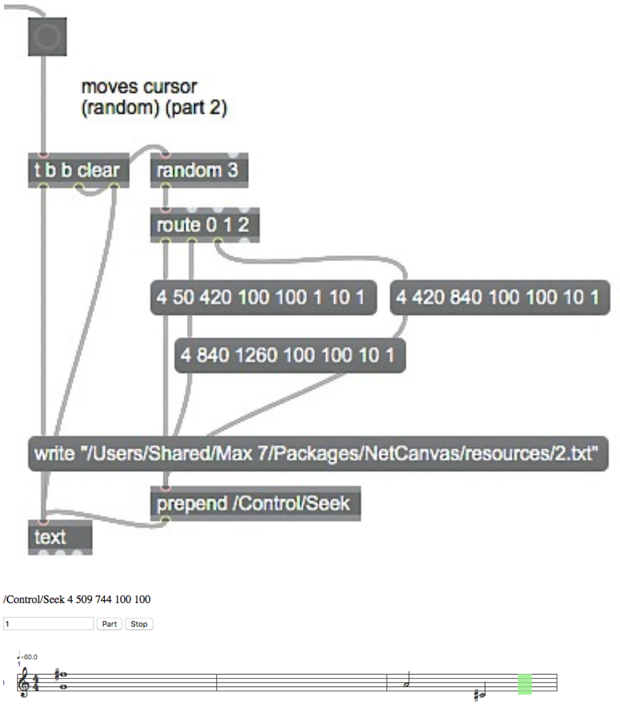


Figure 8. Sending customised cursor instructions from Max to the NetCanvas using the messaging format supported by the Decibel ScorePlayer.

```java
final String textVar = ( UIMain.usersPath + "/" + sessionCounter + ".txt" );
    try {
        File j = new File(textVar);
        FileInputStream bt = new FileInputStream(j);
        ByteArrayOutputStream tout = new ByteArrayOutputStream();
        IO.copy(bt, tout);
            ByteBuffer byteBuffer = ByteBuffer.wrap(tout.toByteArray());
            mSession.getRemote().sendBytes(byteBuffer);
            tout.close();
            System.out.println( byteBuffer);
            byteBuffer.clear();
            }
            catch (IOException e) {
                e.printStackTrace();
            }
```

```javascript
function curs1(x) {
// Score duration
x = x.split(" ");
if (x[0] == "/Control/SetDuration") {
    var duration = x[1];
    document.getElementById("message").innerHTML = "Score duration is set to " + duration;
    }

else if (x[0] == "/Control/SetOptions") {
    var options = x[1];
    document.getElementById("message").innerHTML = "Score options are set to  " + options;
    }

else if (x[0] == "/Control/Play") {
    var options = x[1];
    document.getElementById("message").innerHTML = "Play Started";
    }
else if (x[0] == "/Control/Reset") {
    var options = x[1];
    document.getElementById("message").innerHTML = "Score reset to initial state";
    }
else if (x[0] == "/Tick") {
 var tick = x[1];
 document.getElementById("message").innerHTML = "Current Tick value is " + tick;
}
// Seek commands for layers
else if (x[0] == "/Control/Seek") {
document.getElementById("message").innerHTML = (x[0]+ " " + x[1] + " " + x[2] + " " + x[3] + " " + x[4] + " " + x[5])
  var lay = "layer" + x[1];
  var elem = document.getElementById("layer" + x[1]);
  var xpos = x[2];
  var ypos = x[4];
  var id = setInterval(frame, x[6]);
  var id2 = setInterval(frame, x[6]);
function frame() {
    if (xpos == x[3]) {
      clearInterval(id);
    } else {
      xpos++;
      elem.style.left = xpos + 'px';
    }
    if (ypos == x[5]) {
      clearInterval(id2);
    } else {
      ypos++
      elem.style.top = ypos + 'px';
    }
    }
  }
}
```

Figure 9. Changes made to the NetCanvas Server (above) and the NetCanvas (below) to support the new messaging format and cursor animation.

This involved some changes to the server code, and the NetCanvas code (Figure 9). Following the logic of the existing server implementation, users build text files containing animation instructions in Max, which are transmitted whenever any changes are made. This technique takes advantage of NetCanvas' use of websockets to push data to the browser clients, without requiring interaction on the client side. This makes for a very scalable system, which now supports real-time notation, animated notation, tablet and smartphone scores, network music performance, VR notation, and is fully cross-platform and accessible from Max, and can be used in conjunction with the Decibel ScorePlayer.

4. CREATIVE WORK

The teams form Hamburg and Perth completed and performed a number of works exploring screenscore-based notation (Table 4.). These works included the investigation of the possibilities of both MaxScore and the Decibel ScorePlayer**,** as well as one work exploiting the possibilities of both platforms.

New Works	
Carey, B.	*Magnetic Visions VI* (2016)
Fu, X.	*I Love Tiffany* (2016)
Hajdu, G.	*Carnage* (2016)
Hope, C.	*Great White* (2016)
Vickery, L.	*abstract clouds of the western skies (2016)*
Vickery, L.	*TECTONIC: Rodinia* (2016)
Adapted works	
Dunscombe, S.	*Westpark* (2012)
Vickery, L and Hope, C.	*Talking Board* (2012)

Table 4. Works developed during the residency.

Vickery's 'abstract clouds of the western skies' (2016) is a nonlinear work in which a 68 measure "source score" is deconstructed into passages and single measures to create a texture that alternates indeterminate juxtapositions with more synchronous linear "composed" passages. It is the second work by the composer in this formal framework (the first was 'Improbable Games' (2010)). During the exchange version for both MaxScore and the Decibel ScorePlayer were created. The Canvas mode is used to define the score in the ScorePlayer, but unlike Dunscombe's 'Westpark' separate "parts" are defined for each of the three performers (see Figure 10). In the Decibel ScorePlayer performers change parts using a swipe up or down gesture on the iPad screen [11].

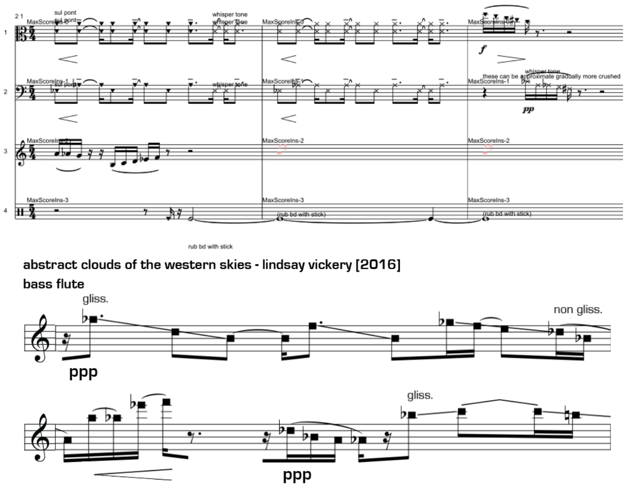


Figure 10. Flute part of Vickery's 'abstract clouds of the western skies' (2016) in MaxScore (above) the Decibel ScorePlayer (below).

The main difference between Quintet.net and the Decibel Score Player lies in how the progress of time is represented. While the ScorePlayer, in its basic mode, moves the score under a static play head, scores in Quintet.net are shown as static pages over which a cursor is moved from left to right. One of the pieces, that originally existed for Quintet.net and therefore had to be adapted to the ScorePlayer's paradigm is Xiao Fu's composition 'Tiffany' written in 2011 for a flexible ensemble of musicians and first performed as part of the Pentalocus network concert between San Diego, Montreal, New York, Belfast and Hamburg in November 2011. Her program notes state:

> *Works inspired by the Tiffany glass. I used the five logos of the participating school and split them into five parts, each part is processed, and then rearranged, in order to finally generate a colorful "Tiffany glass Score".*
> *The work's full-length is five minutes, each page is one minute. From left to right is time, from top to bottom is the approximate pitch range.*
> *If there is only one musical instrument in a given location, different colors represent different playing techniques; or if there is a duo or trio, different colors represent a different instrument.*
> *Each grid block can be a note, but the players are free to choose among simultaneous blocks and can even jump between them as long as they have the same color. White spaces are silence.*
> *The height of each small block, represents dynamics, the higher the box the louder.*

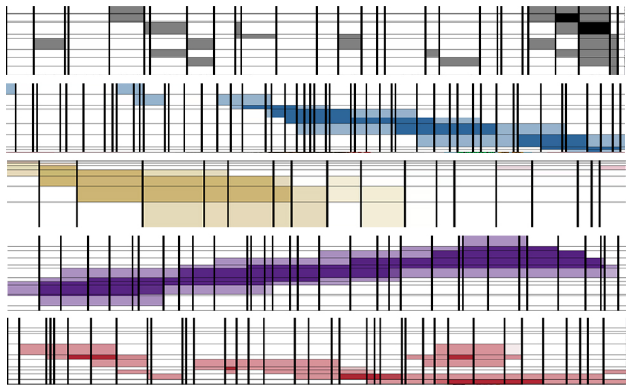


Figure 11. A page from Xiao Fu's composition 'I Love Tiffany' (2016) for a networked music ensemble.

The only work to explicitly explore communication between the Decibel ScorePlayer and Quintet.net at this point in the exchange is Cat Hope's 'Great White' (2016). This work was scored for accordion and viola (or for any two sustaining instruments) reading from the Decibel ScorePlayer and two laptop performers connected to Quintet.net.

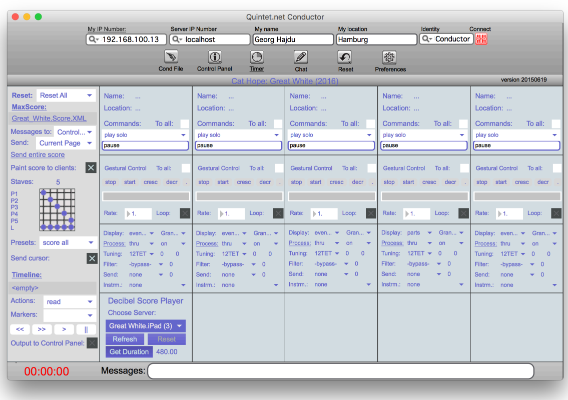


Figure 12. Screen shot of Quintet.net Conductor with built-in control of the Decibel Score Player.

Since the Quintet.net Conductor component—via an instance of the MaxScore Editor—serves the scores to the clients, we implemented a sub-patch capable of communicating with the instance of the Decibel ScorePlayer acting as a server (see Figure 12). After a successful handshake, the Quintet.net Conductor starts the performance by sending the /Control/Play message to the server, upon which it receives a constant stream of ticks which it can sync its own clock to. In this way activity scored for Quintet.net was able to synchronise with the scrolling Decibel ScorePlayer score and therefore the acoustic instruments. This activity is indicated by four sections of traditional notations that appear in blocks on the score. The traditional notations are excerpts of works from composers who had spent some time in Hamburg: including György Ligeti, Gustav Mahler, Georg Telemann and Alfred Schnittke.

The performers of Quintet.net are provided with MIDI files of these excerpts, however certain notes have been removed by the composer beforehand. They are free to interpret these files in any way they wish for the performance, but can only do that when indicated in the score.

The title 'Great White' refers to the great white dead men of music history and women composer's struggle to find a place in that history. It is also a reference to the rare species of (occasionally "man-eating") shark that is hunted in the composers' home state of Western Australia. The acoustic instruments interface with the music excepts (already being altered by the electronic performers) firstly by reading the graphic score that 'sidesteps' the excerpt, then tracing along the next one, then increasingly 'interfering' with the remaining excerpts - eventually blocking out most of the excerpt altogether, concluding with general confusion and a 'free for all' to close. The Quintet.net performers perform with the midi files – try as they might, they cannot reproduce the masterworks as they are notated on the score that they follow.

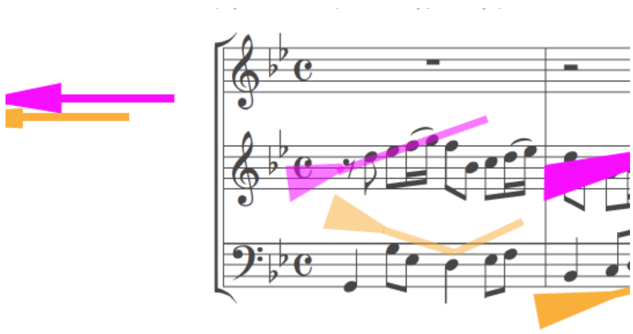


Figure 13. Score showing 'tracing', and some blocking, of the third music excerpt in Cat Hope's 'Great White'.

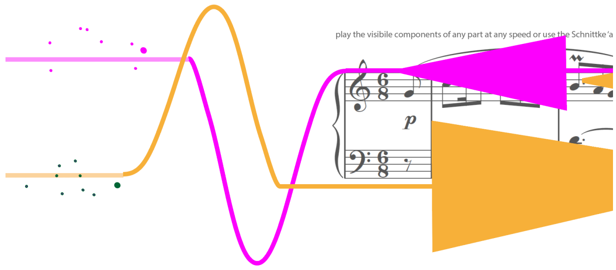


Figure 14. Score showing more 'blocking' of the parts.

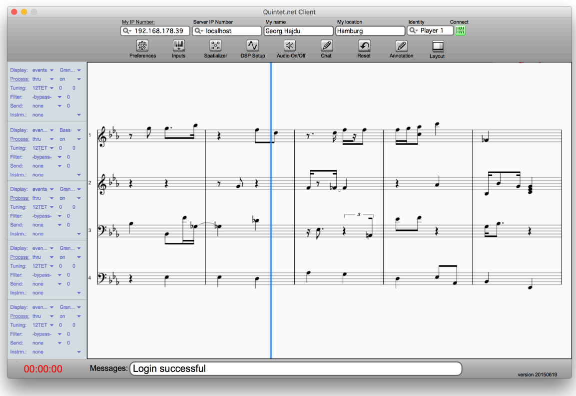


Figure 15. The version of the score for 'Great White' served by the Conductor to the Quintet.net Clients. The cursor is sync'ed to the ticks sent by the Decibel Score Player.

5. CONCLUSIONS

This research collaboration has allowed for some progress in allowing for interoperability between Quintet.net, the Decibel ScorePlayer, MaxScore. Future research will focus on extending the Canvas mode in the ScorePlayer to allow for drawing geometric shapes (and animated notation) using Core Graphics and Core Animation. It is hoped that through this collaborative research, some standardization can be established for communication between various digital score solutions, and that for the end user, there may be some flexibility in the way in which scores are delivered on screen. Future development may also evaluate whether *Ableton Link* – which is fast becoming a de facto standard for synchronising devices to *Ableton Live* over local networks – is a worthwhile means of synchronising various digital score solutions. Although this approach, which has been a key feature of this collaboration thus far, is not designed to function over extended distances, a custom synchronisation method could also be developed or simply adapted from existing solutions.

Acknowledgments

The exchange between the Hochschule fur Musik und Theater (HfMT Hamburg) and Edith Cowan University (Perth) was funded by the Deutscher Akademischer Austauschdienst (DAAD) and Universities Australia.

6. REFERENCES

[1] J, Freeman. "Extreme Sight-Reading, Mediated Expression, and Audience Participation: Real-Time Music Notation in Live Performance," *Computer Music Journal*, 32(3), 2008, 25–41.

[2] C. Hope, A. Wyatt, & L. Vickery. "The Decibel Score Player – A Digital Tool for reading Graphic Notation." *Tenor Conference*, Paris, France, 2015.

[3] G. Hajdu, "Quintet.net: An environment for composing and performing music on the Internet," *Leonardo*, 38, 2005, 23-30.

[4] N. Didkovsky and G. Hajdu, "Maxscore: Music notation in Max/Msp," in *Proceedings of the International Computer Music Conference*, 2008, 483-486.

[5] B, Carey, & G, Hajdu. "Netscore: An image server/client package for transmitting notated music to browser and virtual reality interfaces." *Tenor Conference*, Cambridge, United Kingdom, 2016.

[6] I. I. Bukvic, & J-S Kim. "µ Max-Unity3D Interoperability Toolkit.*" Proceedings of the 2009 International Computer Music Conference*, Montreal, Canada, 2009.

[7] C. Agon, G. Assayag, M. Laurson, and C. Rueda. "Computer Assisted. Composition at Ircam: PatchWork & OpenMusic." *Computer Music Journal* 23(5), 1999, 59-72.

[8] S. Shafer. "Performance Practice of Real-Time Notation." *Tenor Conference*. Cambridge, United Kingdom, 2016.

[9] A. Agostini, & D. Ghisi. "Gestures, Events, and Symbols in the Bach Environment." *Actes des Journées d'Informatique Musicale* (JIM 2012), Belgium, 2012.

[10] D. Fauber, Y. Orlarey, S. Letz, "INScore - An Environment for the Design of Live Music Scores", Proceedings of the Linux Audio Conference, LAC 2012.

[11] D. Steinberg. *Zero Configuration Networking: The Definitive Guide*. O'Reilly Media, 2005.

[12] A. Wyatt, C. Hope, L. Vickery, & S. James. "Animated Music Notation on the iPad," In *Proc. Int. Computer Music Conf. 2013*, Perth, Western Australia, 2013, 201-207.

[13] C. Hope, A. Wyatt, & L. Vickery. "The Decibel Scoreplayer: Enriched Scores for the iPad," in *Proc. Int. Computer Music Conf,* 2015.

[14] L. Vickery, & S. James. "Tectonic: a networked, generative and interactive, conducting environment for iPad," in *Proc. Int. Computer Music Conf. 2016*, Utrecht, The Netherlands, 2016, 542-547.

[15] K. Yarmosh. *App Savvy: Turning Ideas into iPad and iPhone Apps Customers Really Want.* O'Reilly Media, 2010, p. 53.

[16] L. Vickery, C. Hope, & S. James. "Digital adaptions of the scores for Cage Variations I, II and III," in *Proc. Int. Computer Music Conf. 2012,* Ljubljana, Slovenia, 2012, 426-432.

APPENDIX

List of Commands for the Decibel ScorePlayer.

Note: For externals, only the following classes of commands are forwarded to the iPad clients.

/Control /Renderer /Master
And only the following message classes are forwarded to externals.

/External (This is not broadcast to iPad clients.)

/Control /Status /Tick
The following commands are for client server interaction, and form part of the initial handshake.

/Server/ProtocolVersion *versionString*
Sent by the server to the client on initial connection. The version string reflects the protocol version used by the server and is of the format "Decibel Networking Protocol v13".

/Server/RegisterDevice *versionString identifier playerVersion*
Once the client receives the /Server/ProtocolVersion message, it needs to respond to the server with this message. The version string should match the one given by the server. If not, the player should abandon the connection and alert the user. The identifier is the hostname of the device. (Some rendering modules give you the option of specifying the identifier. This is mostly for interactions with externals.) The playerVersion is the version of the app itself. (At the time of this writing, the current development version was 1.6.4)

/Server/ConnectionOK
If the /Server/RegisterDevice command was successful, the server will notify the client with the above message.

/Server/MakeSecondary
Once the handshake is completed, the server sends this command to the client if there is not already a backup server on the network. The client then sets itself up as a secondary server and informs the primary server of the port number used for it.

/Server/SecondaryPort *portNumber*
The return command sent upon the creation of a secondary server, including the listening TCP port number.

/Server/CancelSecondary
If the primary server doesn't get a response from the secondary server within a 3 second timeout period then it elects a new client as a secondary (if there are other clients on the network) and adds the original secondary to a blacklist. It also sends this command to the failed secondary to let it know that its services are no longer required.

/Server/SecondaryServer *address port*
Once a backup server has successfully been established, the primary server sends this message to all of the connected clients. The clients should then connect to the secondary server using the same handshake. If the primary server goes down then the secondary server seamlessly takes over.

/Server/RegisterExternal *versionString udpPortNumber*
This command is used by an external (for example a Max patch) connecting to the server over UDP. The versionString should match the one used to establish a normal client/server connection, and the udpPortNumber should be a port that the external is listening on for return messages.

/Server/RegisteredExternals *address1 port1 address2 port2...*
The /Server/RegisteredExternals message is sent from the primary server to the secondary server so that it can maintain an up to date list of externals in the case that it is called upon to replace the primary server. Multiple externals can be specified in the argument list.

/External/NewServer *hostName udpPortNumber*
If the secondary server takes over, it sends this command to all of the registered externals to notify them that the server address has changed. The external should then re-register with the secondary server. (The secondary server clears the list of registered externals after sending this message, taking the server change as an opportunity to remove any dead externals from the list.)

/Server/GetClientList
Sent from a client to the server to get a list of all of the connected clients. (Used by the NetworkViewController to show the state of the network.)

/Server/ClientList *identifier1 playerVersion1 identifier2 playerVersion2...*
After receiving a client list request, the server responds with this message. The arguments are a list of the identifiers (usually the hostnames) and the player versions of the connected devices.

/Malformed
This special message is sent internally within the server module to alert it to the presence of a malformed message within an OSC bundle. It is currently quietly ignored, as is the malformed message.
These next commands are more general control commands.

/Control/Play
Begins playback from the current location.

/Control/Reset
Resets the ScorePlayer. (Stops playback and sets the score back to the starting location.)

/Control/Seek *location*
Seek to the specified location in the score. The location should be a floating point value between 0 and 1. For some scrolling scores, with an instruction area defined to the left of the starting point, this value can be set less than 0, but only while the score is stopped.

/Control/SetDuration *duration*
Sets the duration of the score in seconds. This command should not be sent while the score is playing. (In newer versions of the score player it will be safely ignored if this is the case. In older versions, this could have unexpected results.)

/Control/SetOptions *options...*
Sets options for the current score. The possible options, and the format the argument list should take, are defined by the individual renderer modules.

/Renderer
This class of message is passed by the player to the renderer class. The "Renderer" component of the address is added and stripped by the player to facilitate routing – the renderer itself has no knowledge of its existence.

/Status *scoreName scoreType "Reserved" playerState location duration ("CurrentOptions" options...)*
Returned from the master in response to a /Master/GetStatus request. The scoreName and scoreType arguments are used to verify the name of the score and the rendering class used by it. (These were added in Protocol version 12 when manual network connections were first allowed. Zeroconf discovery only finds servers that are running the same score as defined by the name and composer.)
The "Reserved" argument is for the possible future implementation of score version checking. PlayerState shows the current state of the player – 0 for stopped, 1 for playing.
The location is the current location within the score as a float between 0 and 1, and the duration is the current length of the score in seconds. (Important for scores where the duration can be changed.)
The "Current Options" argument is optional, and is only present if the score has options that need to be set. The remaining arguments that follow are options and are dependent on the rendering module used.

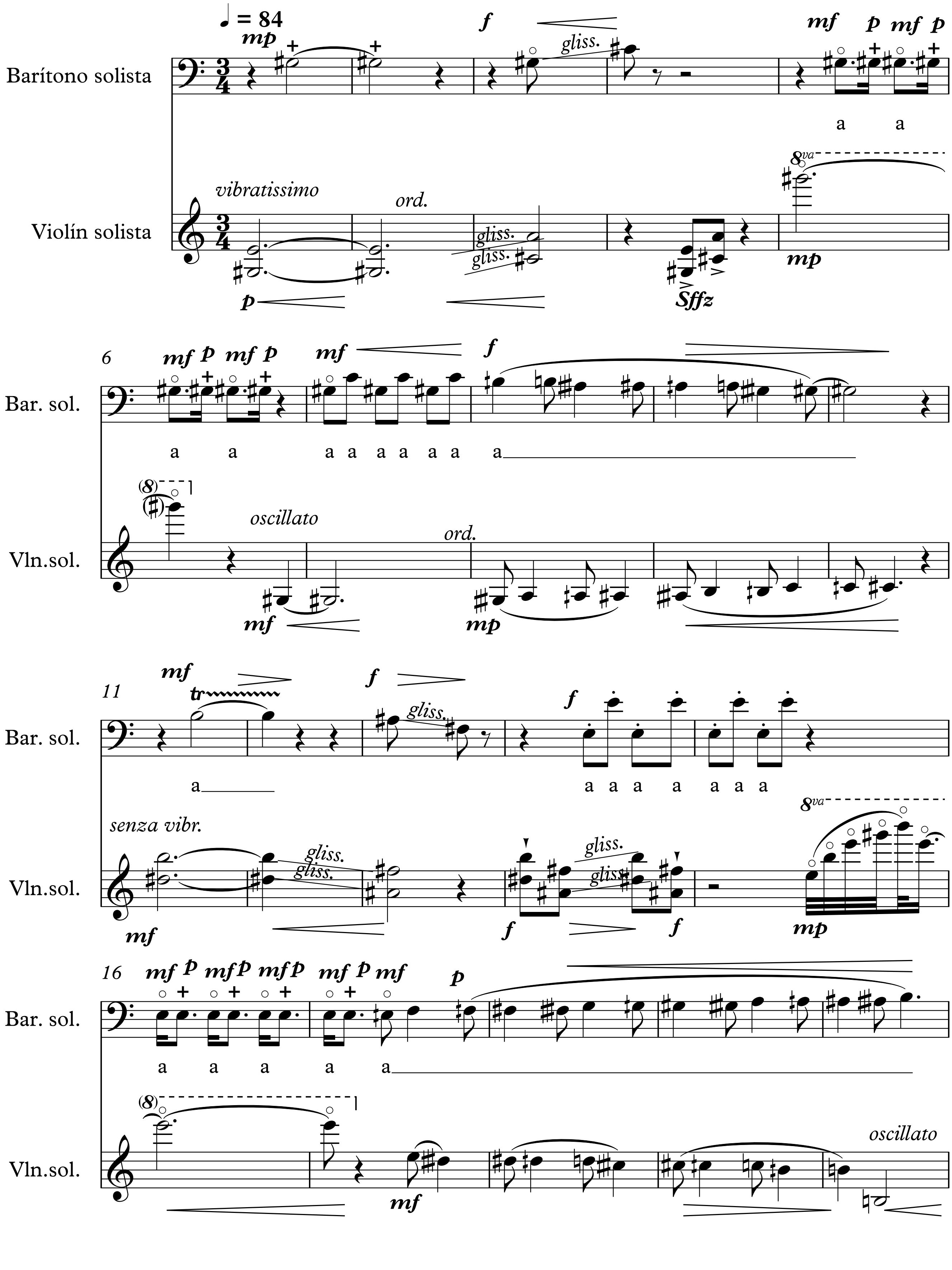
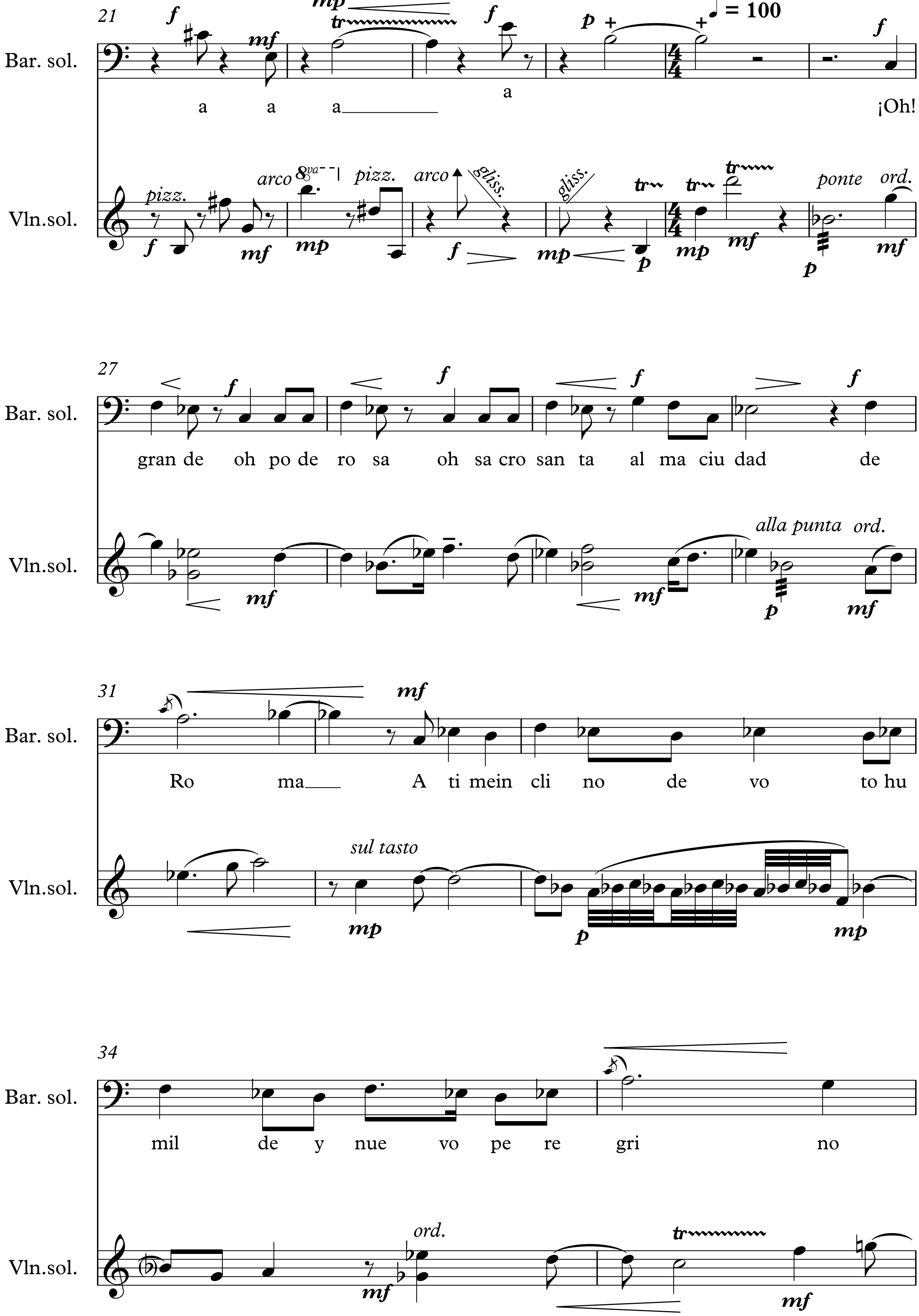
/Tick location
The tick message is sent across the network every second by the master with a float argument between 0 and 1 which represents the current location within the score.

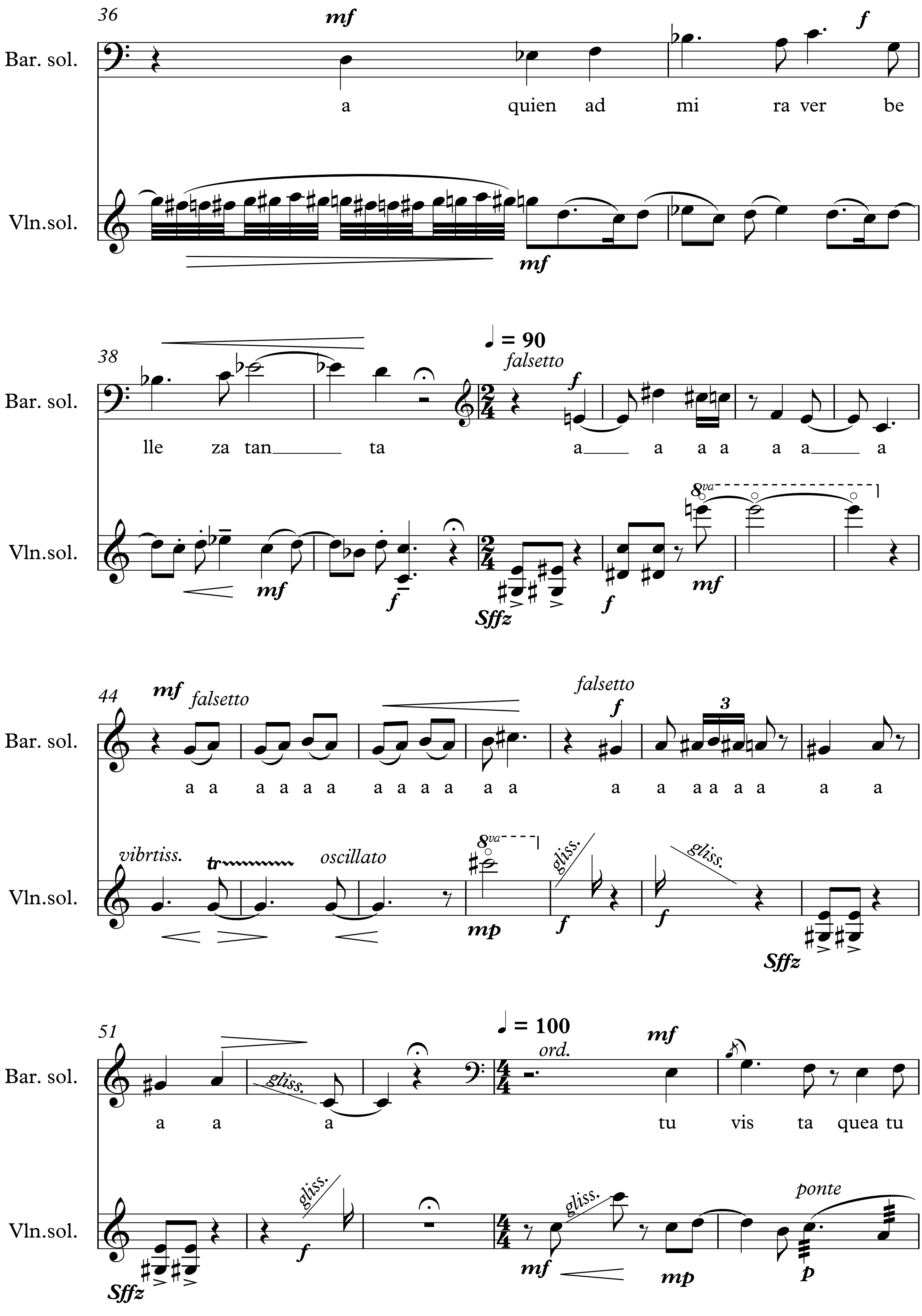
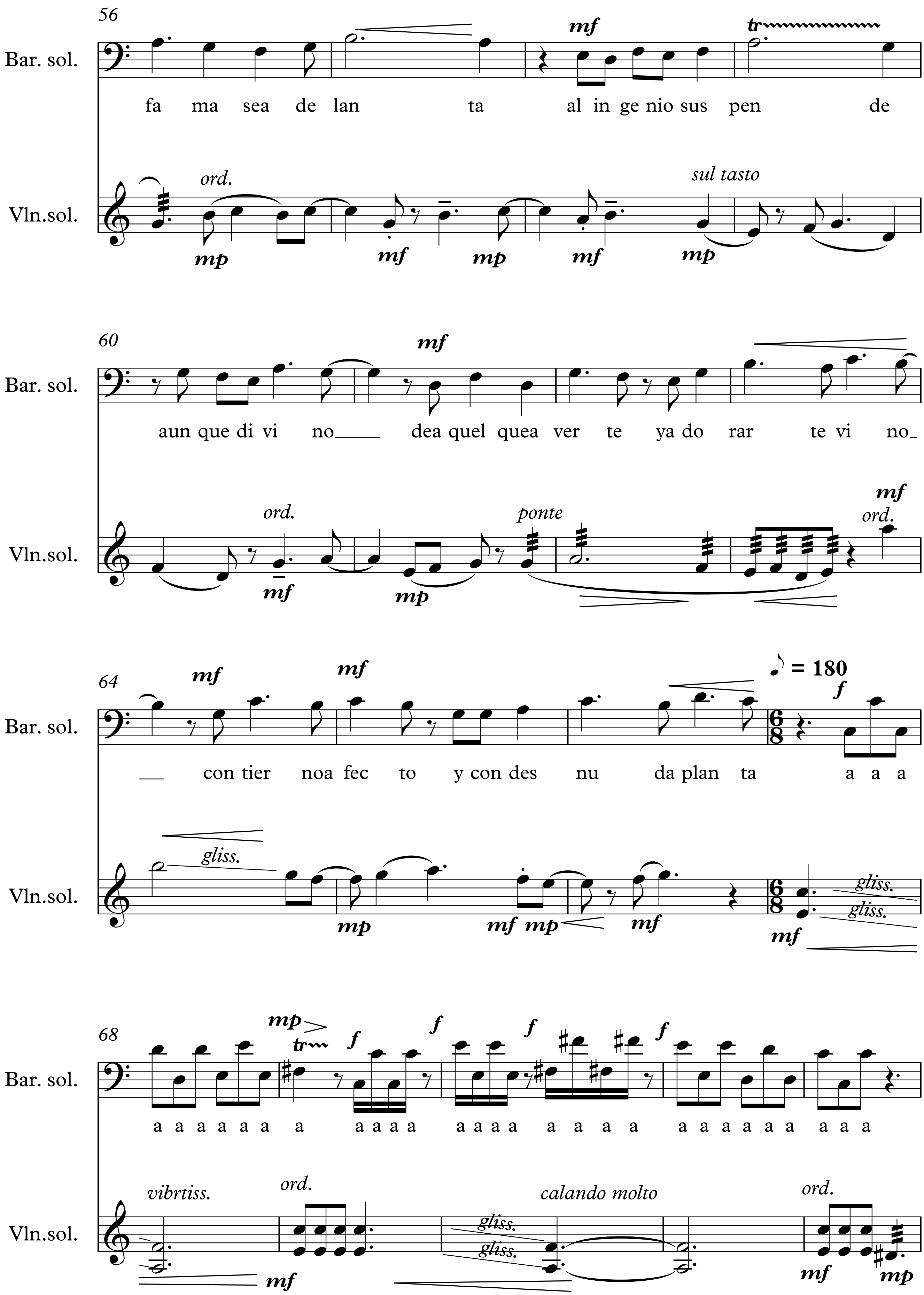
/Master/GetStatus
Sent by clients after completing the handshake with the server to request the current state of the score from the master. (See /Status)

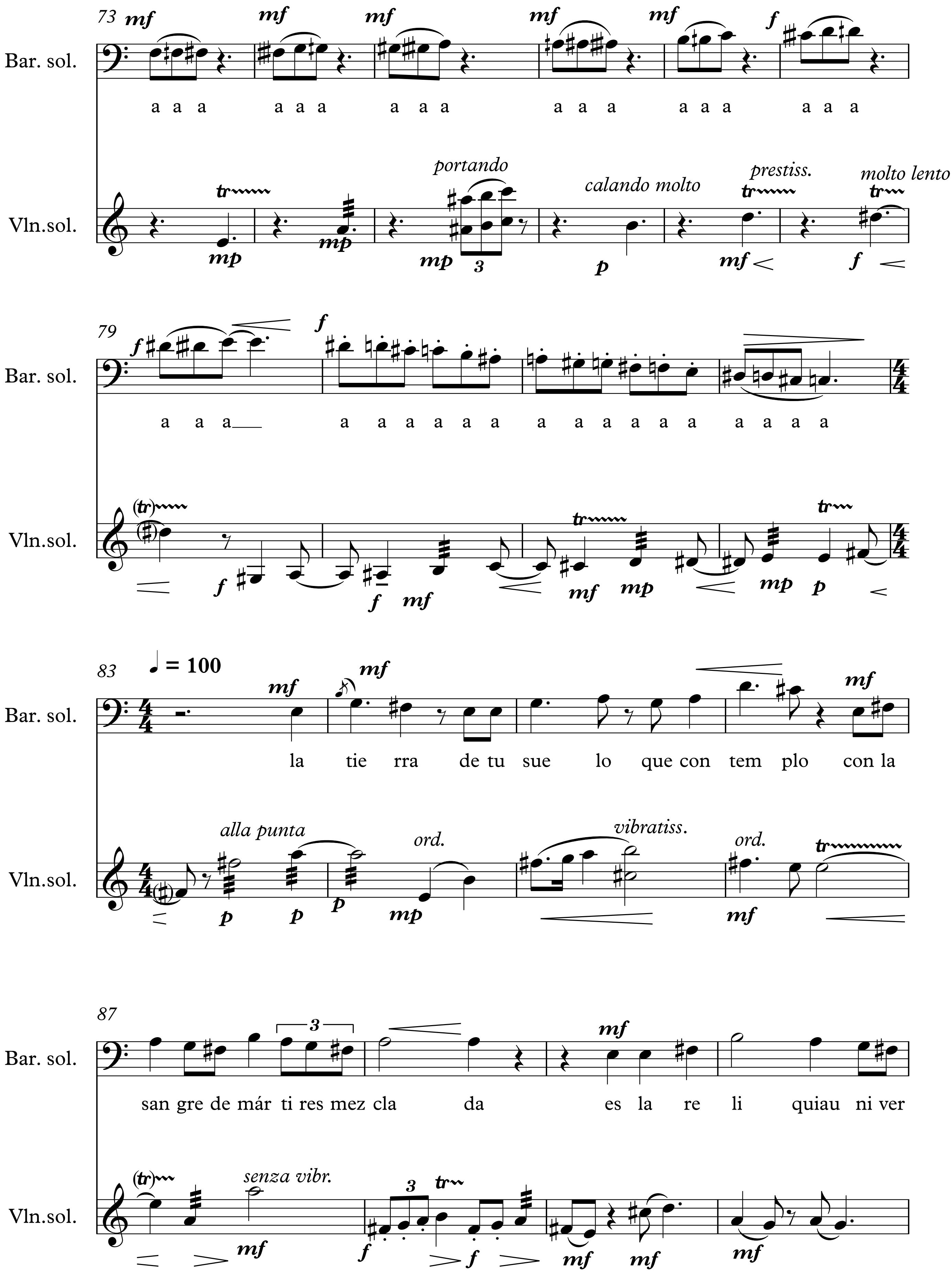
Persiles avista Roma

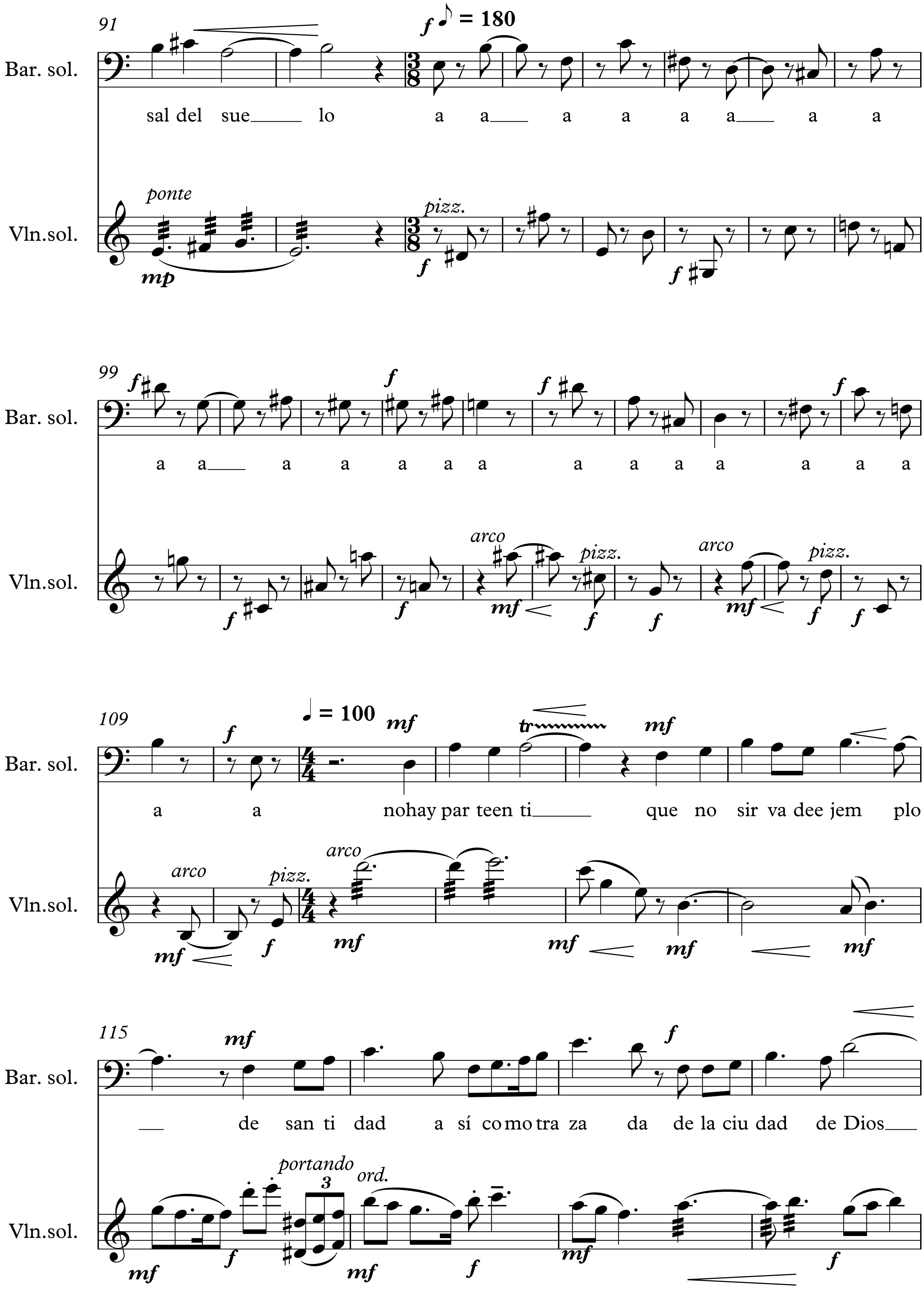
para barítono y violín

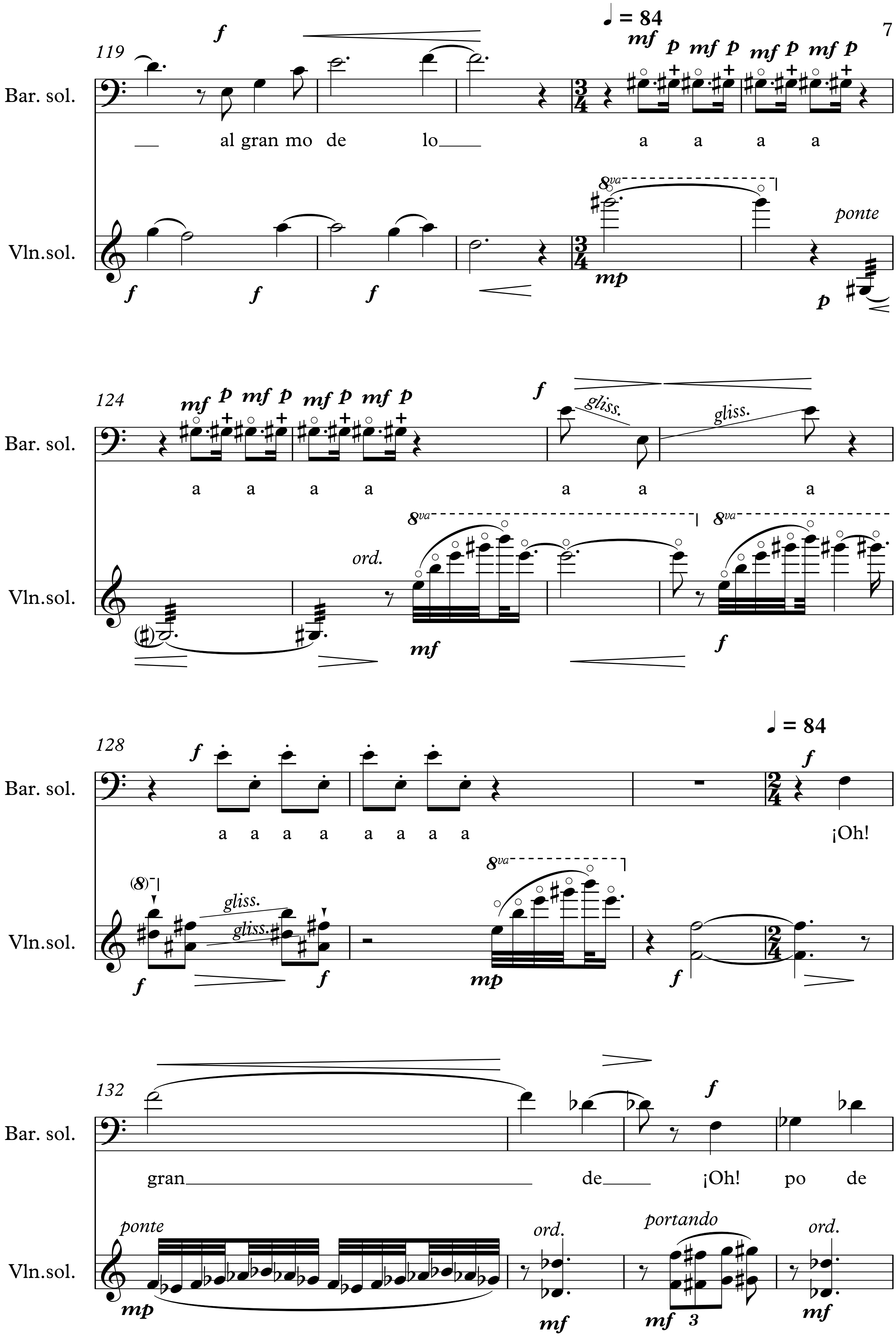
Texto: Miguel de Cervantes

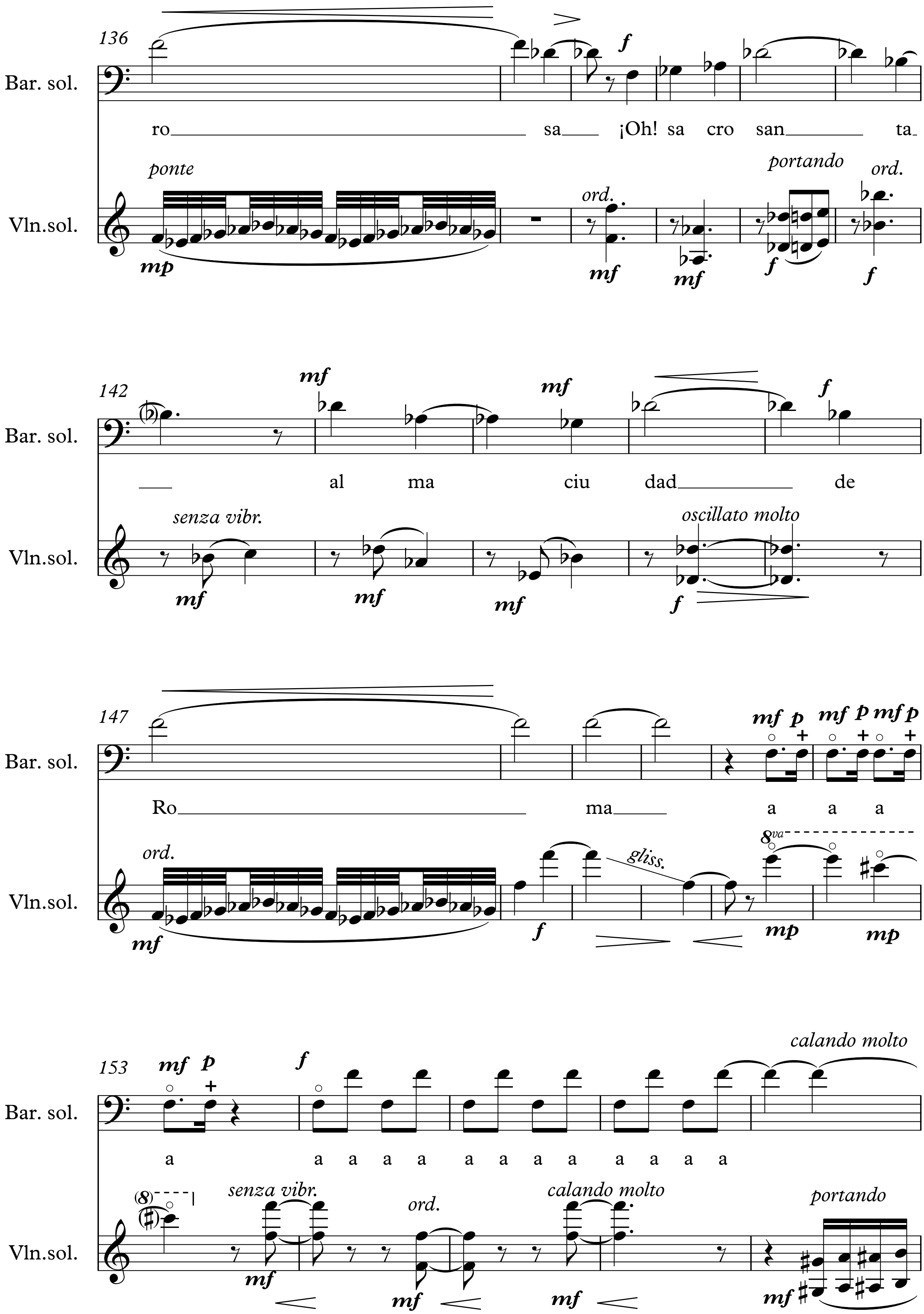
Música:Tomás Marco


Bar. sol.
Vln.sol.
= 100
a a a a ¡Oh!
pizz. arco 8va pizz. arco gliss. gliss. ponte ord.
gran de oh po de ro sa oh sa cro san ta al ma ciu dad de
alla punta ord.
Ro ma A ti mein cli no de vo to hu
sul tasto
mil de y nue vo pe re gri no
ord.

Bar. sol.
Vln.sol.
a quien ad mi ra ver be
lle za tan ta
♩ = 90
falsetto
a a a a a a a
vibrtiss.
oscillato
gliss.
a a a a a a a a a a a a a a a a a a a
a a a
♩ = 100
ord.
tu vis ta quea tu
ponte
Sffz


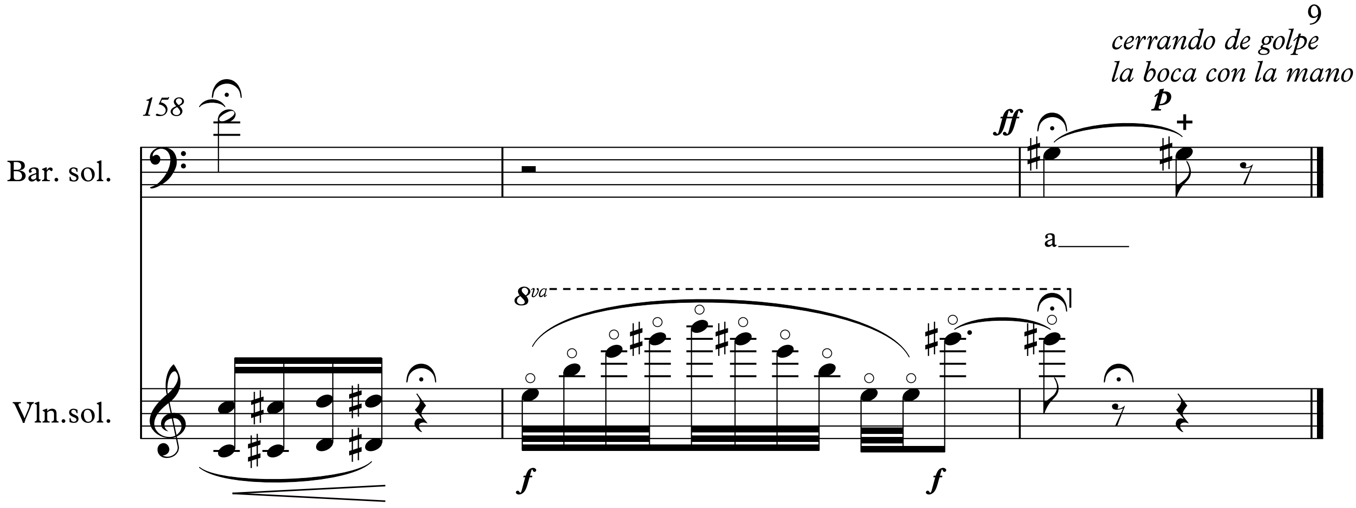
56
Bar. sol.
Vln.sol.
fa ma sea de lan ta al in ge nio sus pen de
ord.
sul tasto
60
aun que di vi no dea quel quea ver te ya do rar te vi no
ord.
ponte
ord.
64
con tier noa fec to y con des nu da plan ta a a a
gliss.
♪ = 180
68
a a a a a a a a a a a a a a a a a a a a a a a a a a a a
vibrtiss.
ord.
calando molto
gliss.
ord.


Bar. sol.
Vln.sol.
a a a a a a a a a a a a a a a a a a
portando
calando molto
prestiss.
molto lento
a a a a a a a a a a a a a a a a a a a a a a
♩ = 100
la tie rra de tu sue lo que con tem plo con la
alla punta
ord.
vibratiss.
ord.
san gre de már ti res mez cla da es la re li quiau ni ver
senza vibr.


91
♪ = 180
Bar. sol.
sal del sue lo a a a a a a a a
Vln.sol.
ponte
mp
pizz.
f
99
a a a a a a a a a a a a a a
arco
pizz.
mf
109
♩ = 100
a a nohay par teen ti que no sir va dee jem plo
tr
arco
pizz.
115
de san ti dad a sí co mo tra za da de la ciu dad de Dios
portando
3
ord.


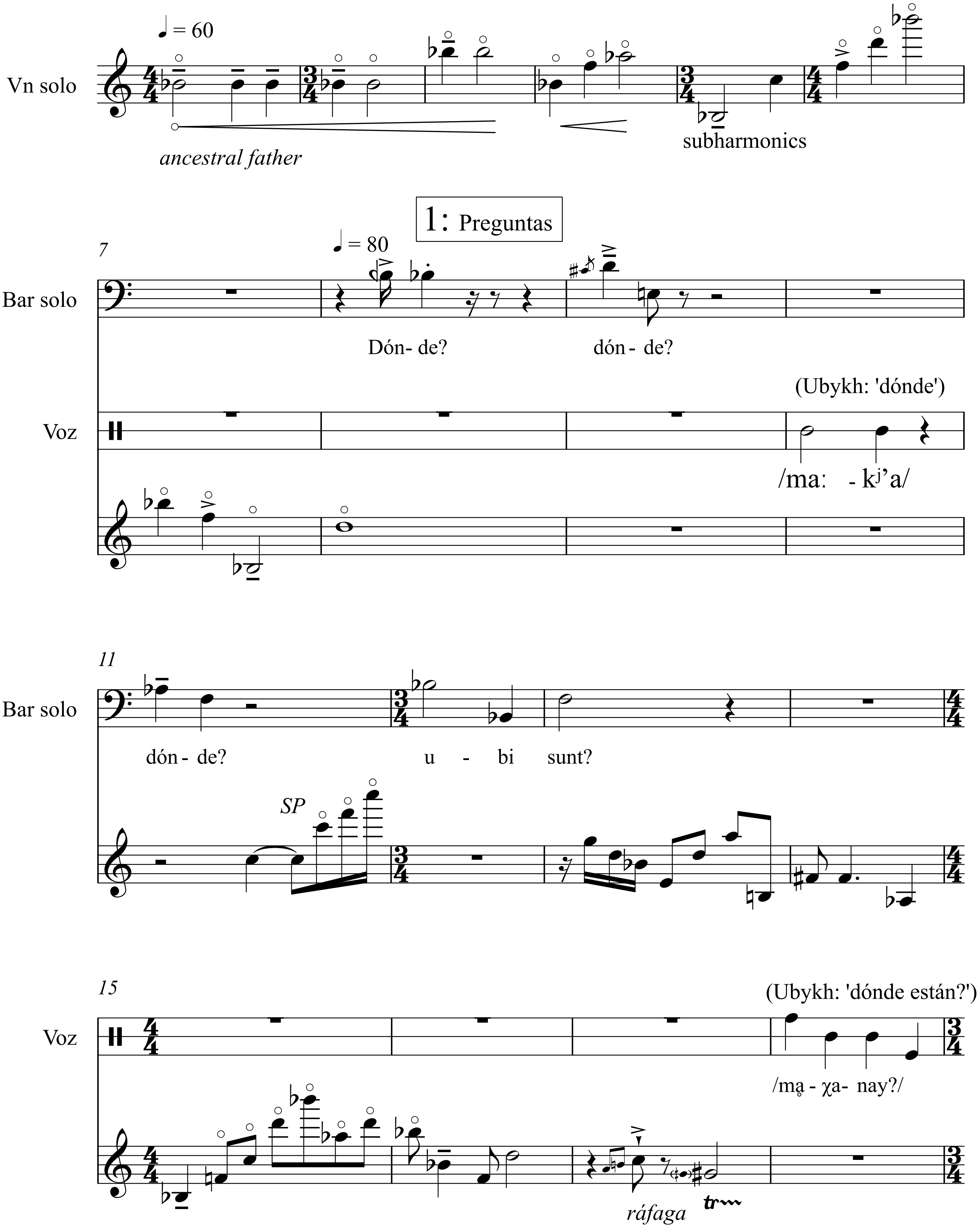
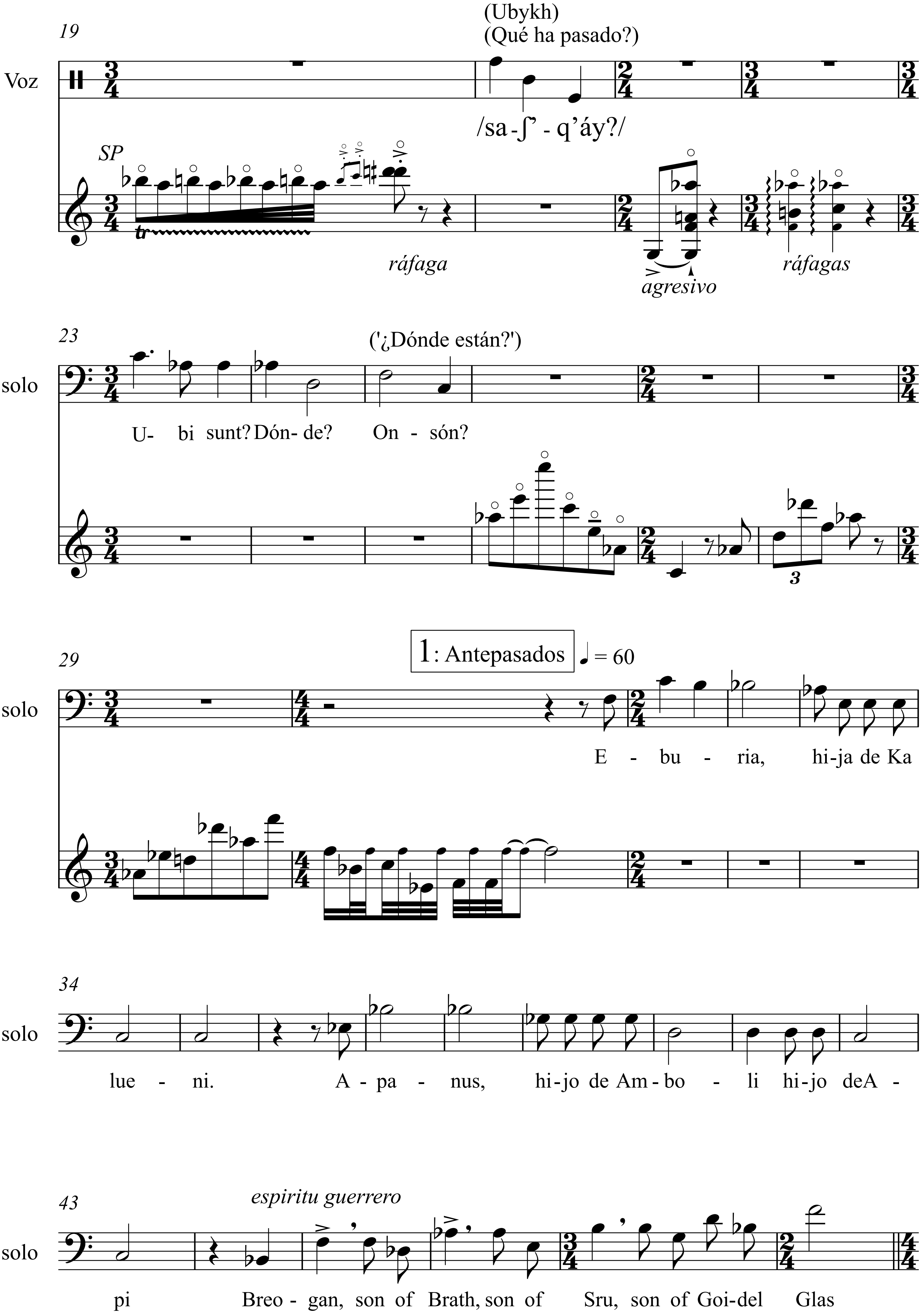
119
Bar. sol.
Vln.sol.
♩ = 84
al gran mo de lo
a a a a
ponte
124
a a a a
gliss.
a a a
ord.
128
a a a a a a a a
gliss.
gliss.
♩ = 84
¡Oh!
132
ponte
gran
de
¡Oh! po de
ord.
portando
ord.


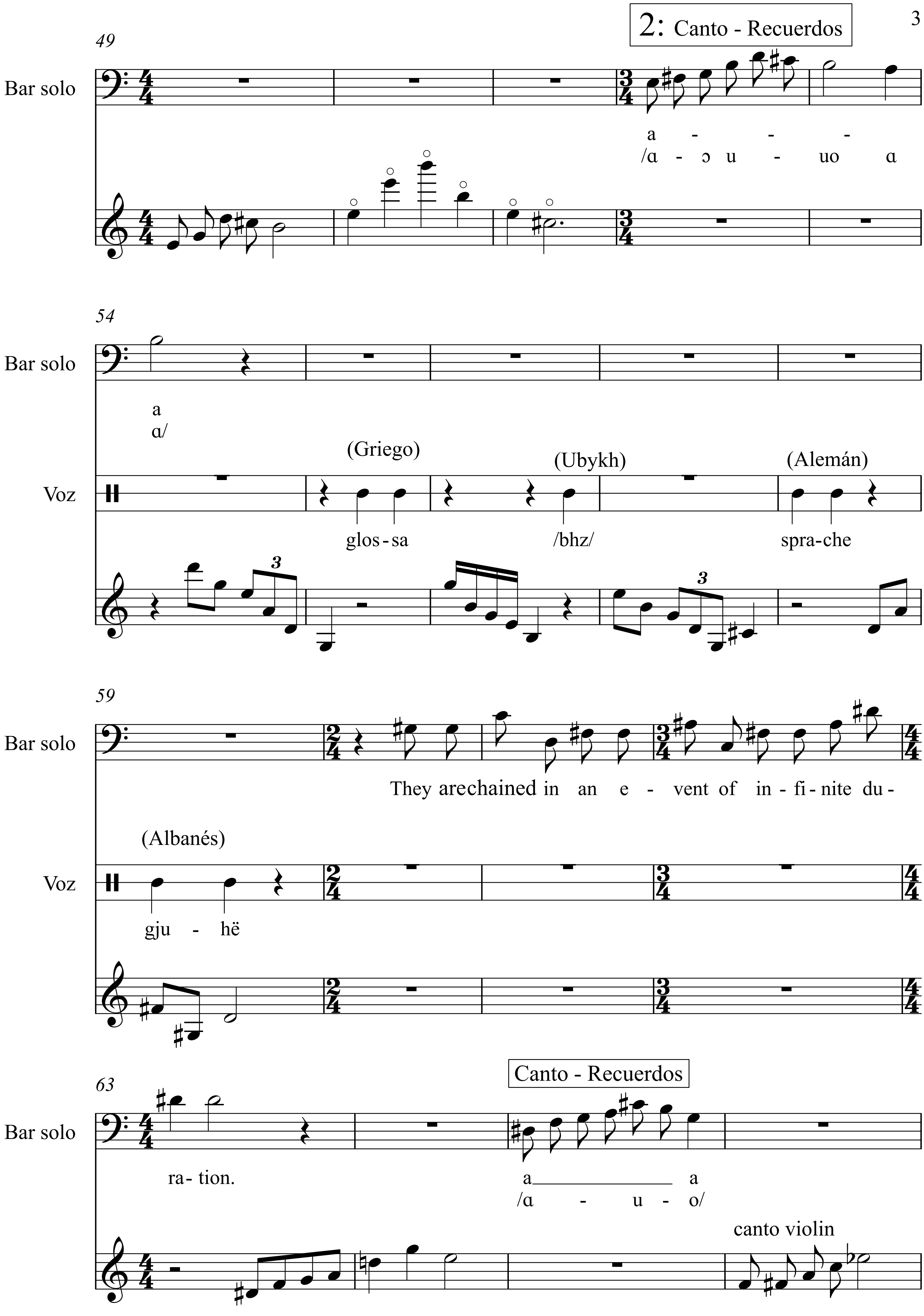
136
Bar. sol.
ro sa ¡Oh! sa cro san ta
Vln.sol.
ponte
mp
ord.
mf
mf
portando
f
ord.
f
142
Bar. sol.
mf
mf
f
al ma ciu dad de
Vln.sol.
senza vibr.
mf
mf
mf
oscillato molto
f
147
Bar. sol.
mf p mf p mf p
Ro ma a a a
Vln.sol.
ord.
mf
f
gliss.
8va
mp
mp
153
Bar. sol.
mf p
f
calando molto
a a a a a a a a a a a a a
Vln.sol.
(8)
senza vibr.
mf
ord.
mf
calando molto
mf
portando
mf


cerrando de golpe
la boca con la mano
158
Bar. sol.
a
Vln.sol.

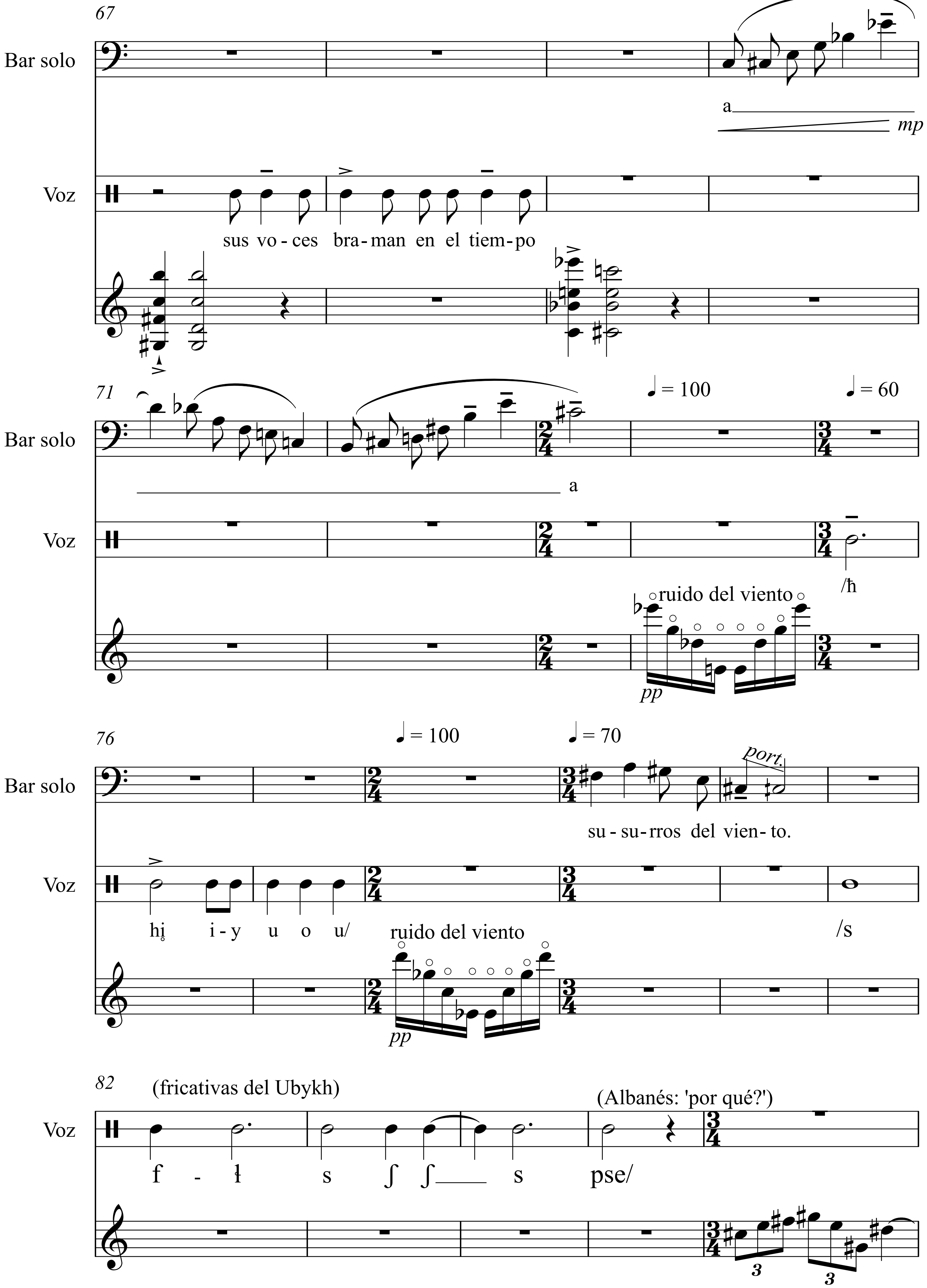
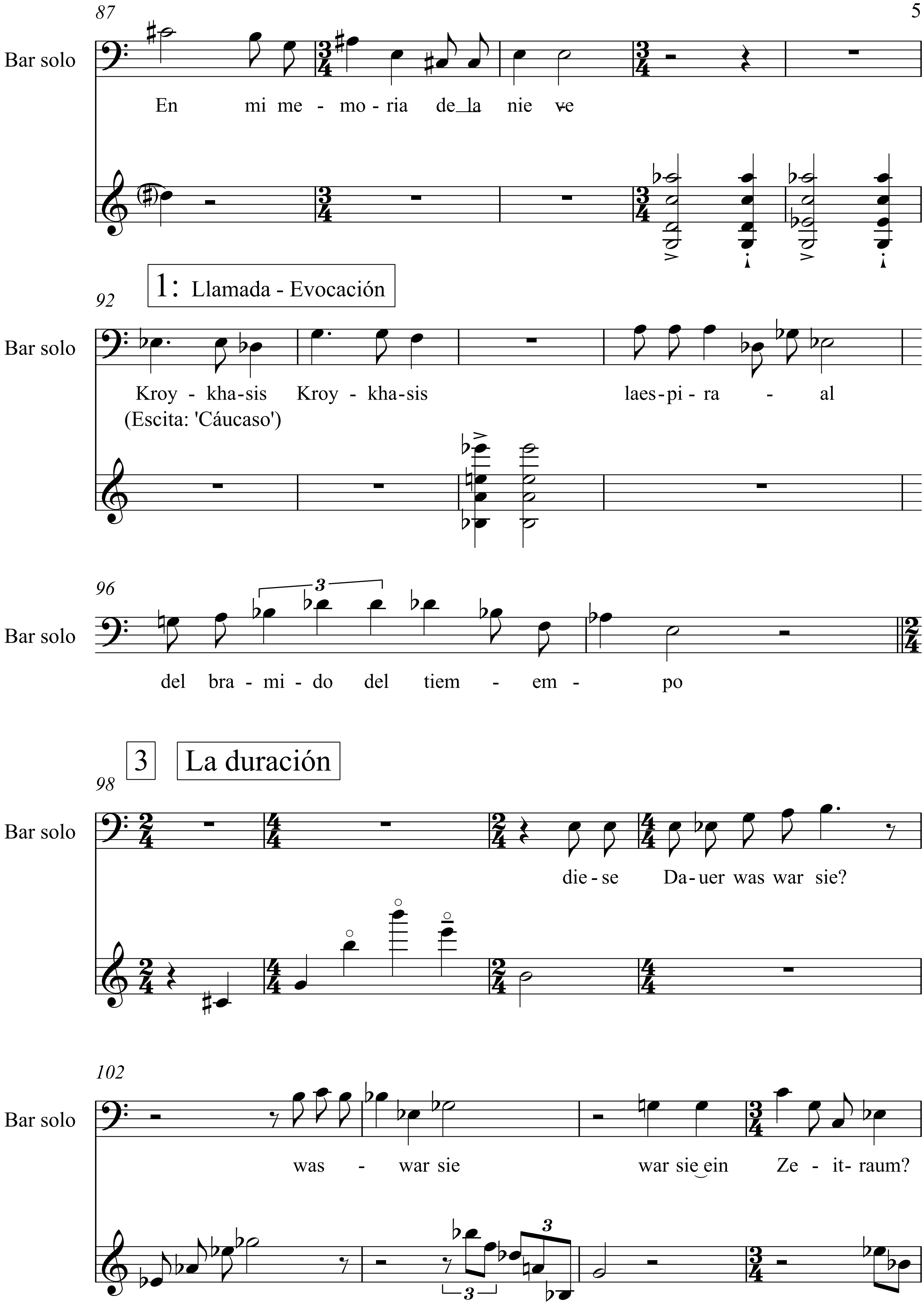
Voice Prints

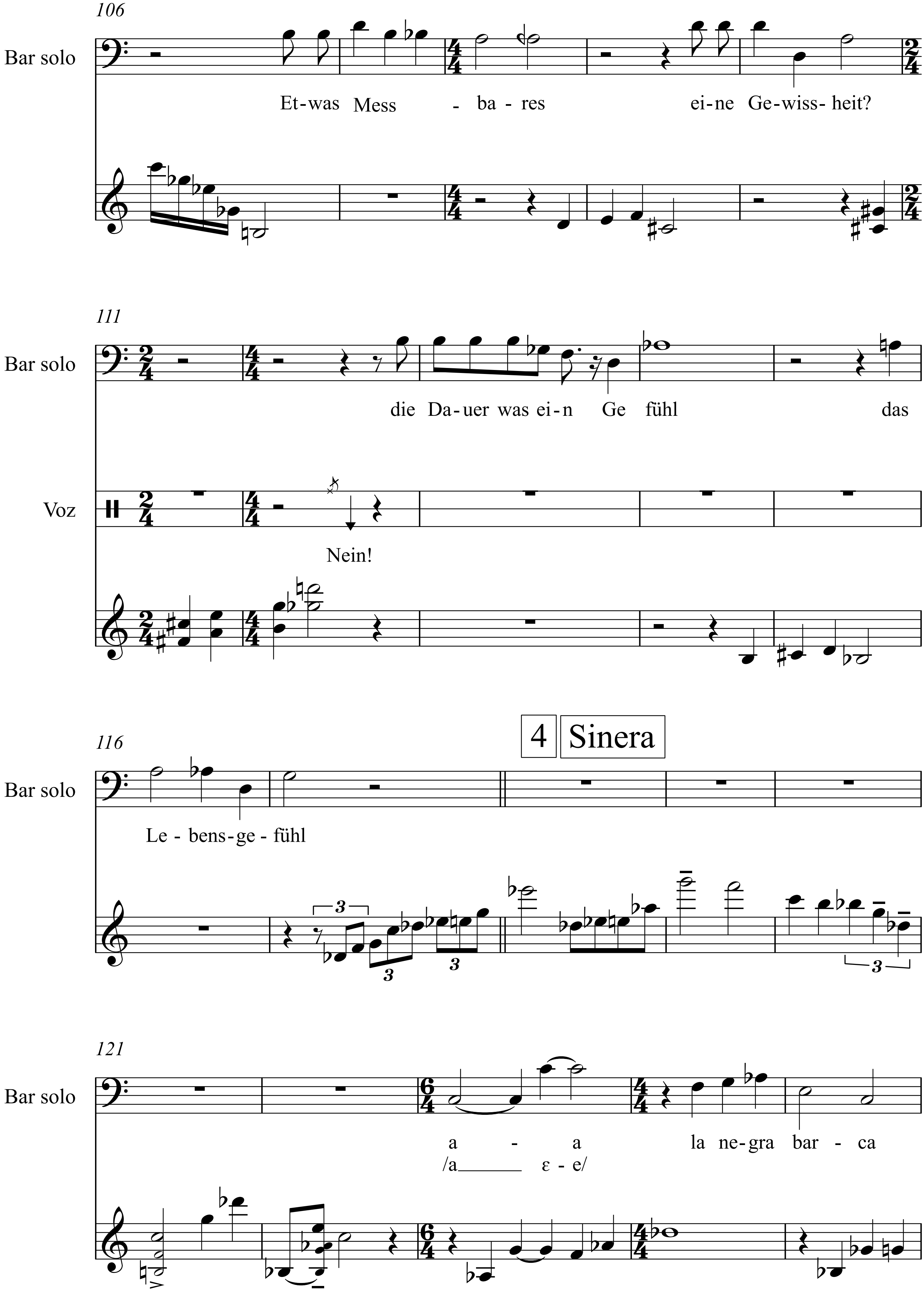
Para Alfredo Garcia y Florian Vlashi

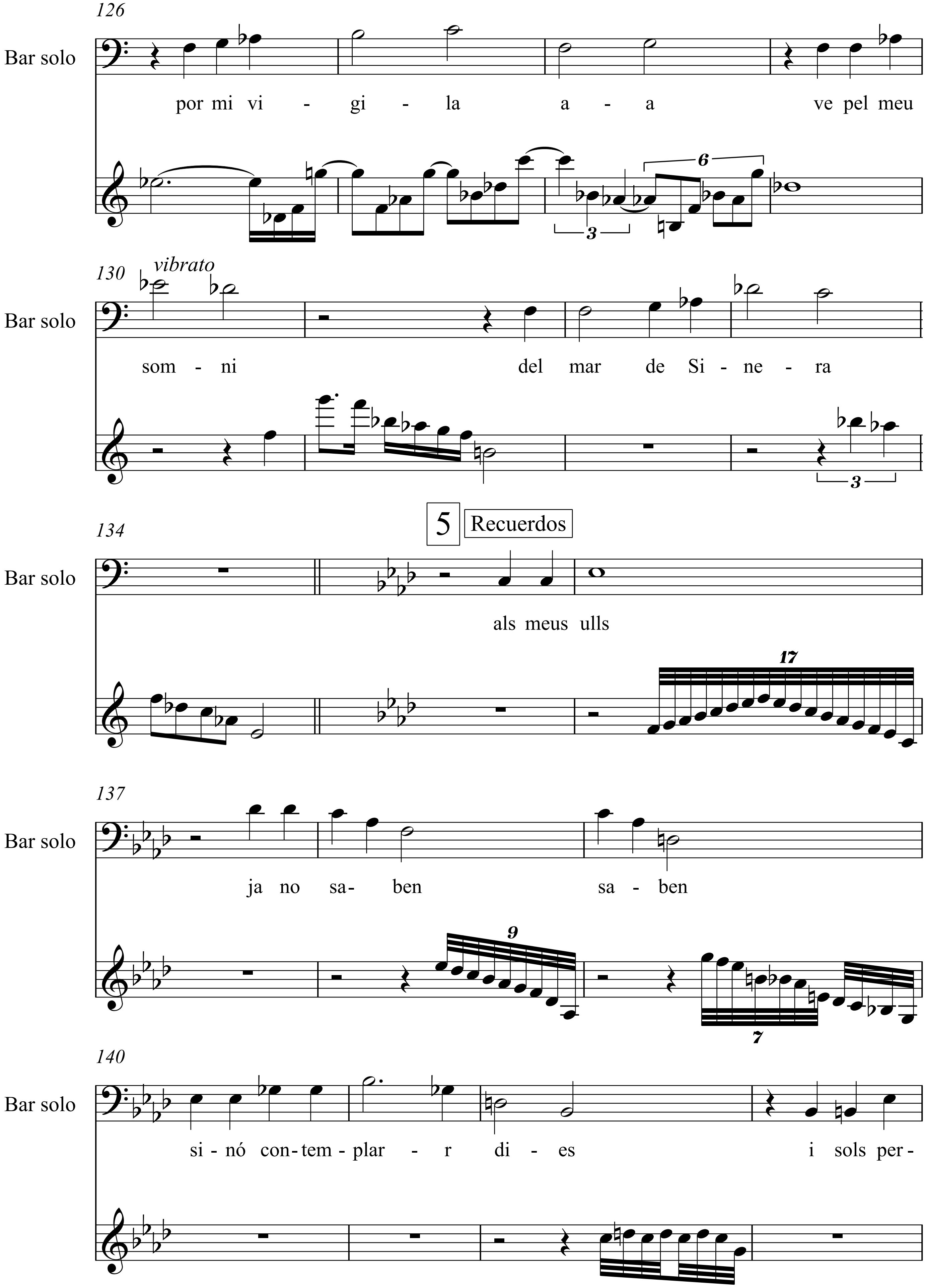
Helena Palma


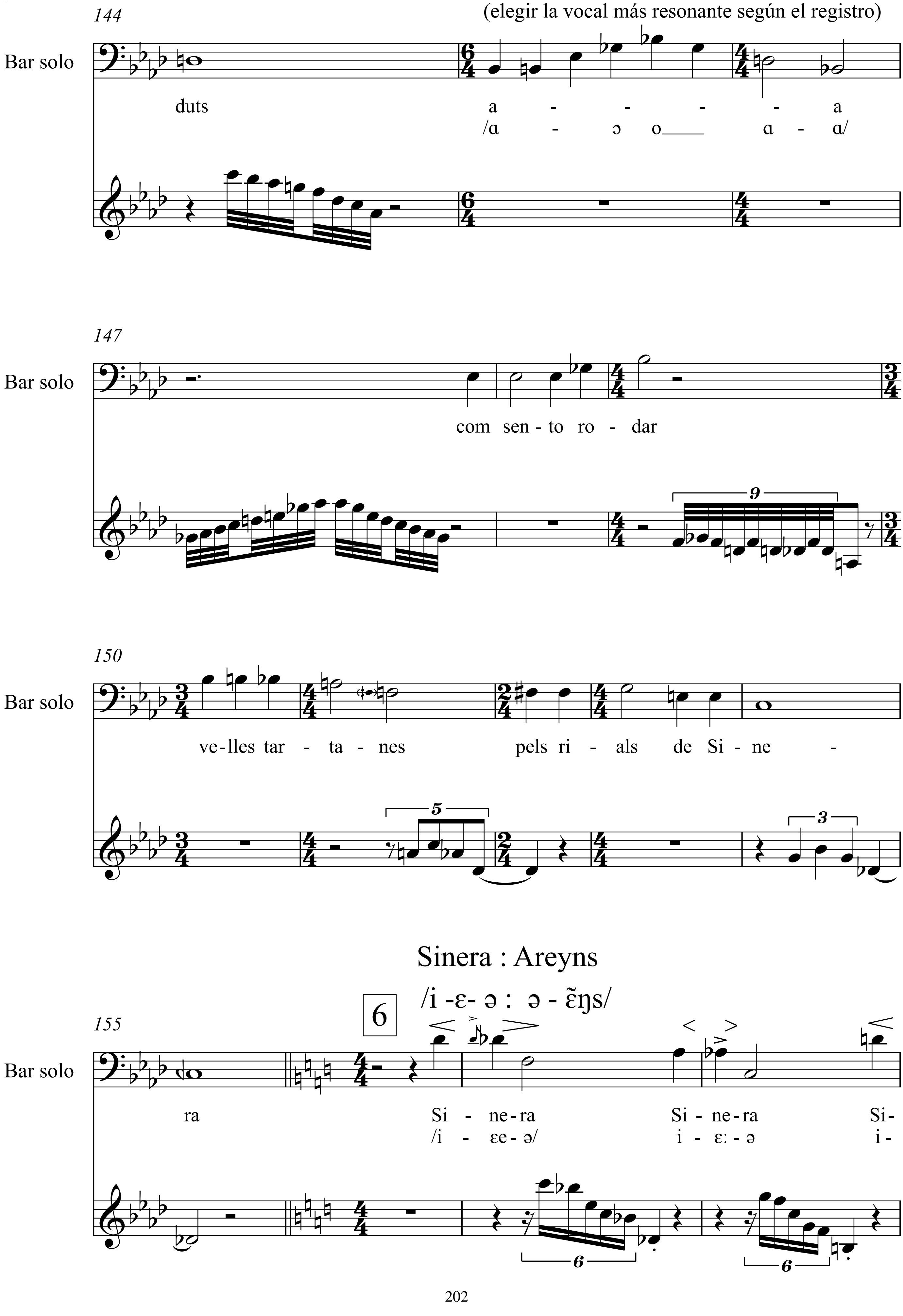
19
(Ubykh)
(Qué ha pasado?)
Voz
/sa - ʕ' - q'áy?/
SP
ráfaga
agresivo
ráfagas
23
('¿Dónde están?')
Bar solo
U- bi sunt? Dón- de? On - són?
3
29
1: Antepasados
♩ = 60
Bar solo
E - bu - ria, hi-ja de Ka
34
Bar solo
lue - ni. A - pa - nus, hi-jo de Am - bo - li hi-jo deA -
43
espiritu guerrero
Bar solo
pi Breo - gan, son of Brath, son of Sru, son of Goi-del Glas


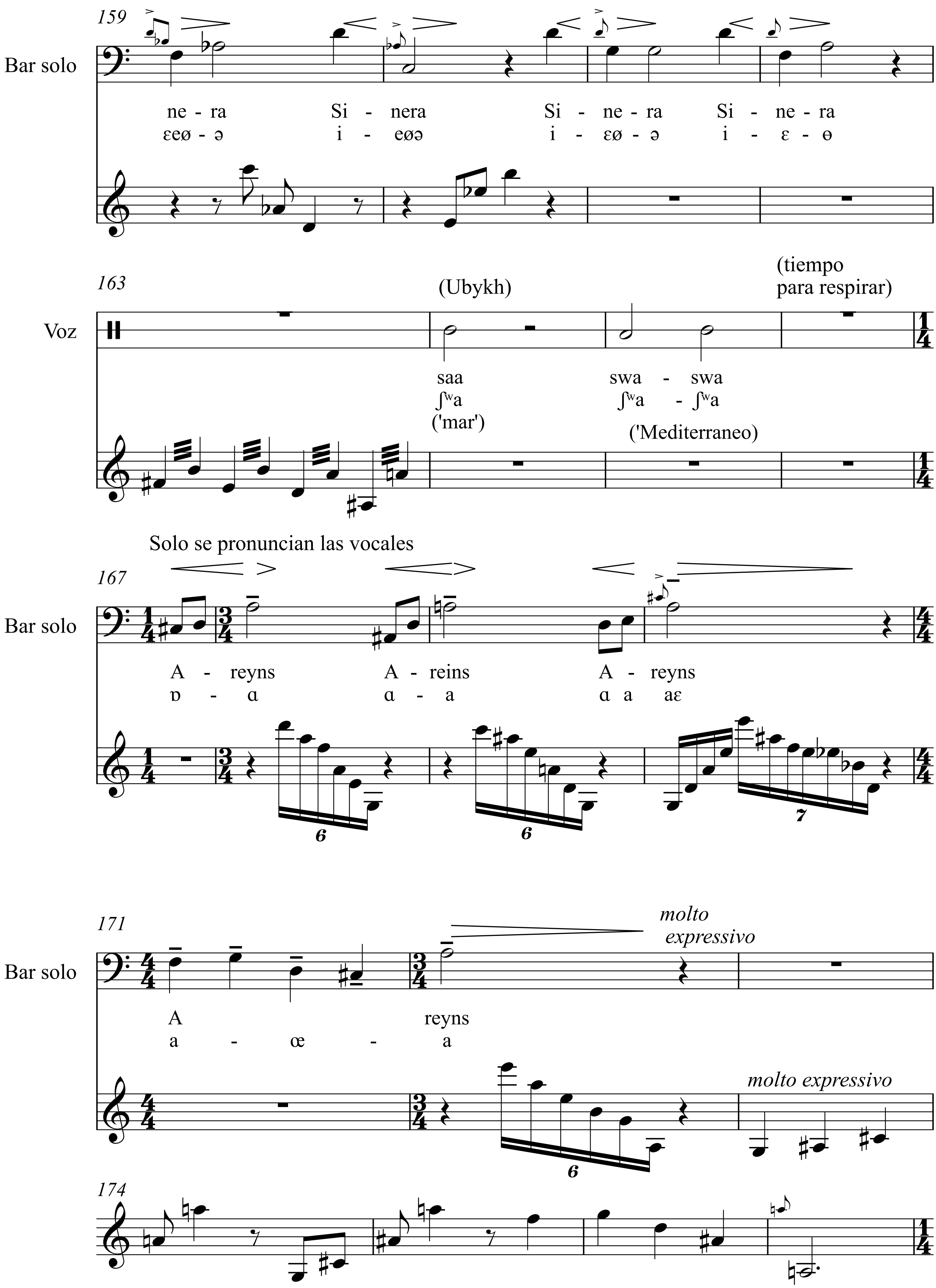
2: Canto - Recuerdos
49
Bar solo
a - - -
/ɑ - ɔ u - uo ɑ
54
Bar solo
a
ɑ/
(Griego)
(Ubykh)
(Alemán)
Voz
glos - sa
/bhz/
spra - che
3
3
59
Bar solo
They are chained in an e - vent of in - fi - nite du -
(Albanés)
Voz
gju - hë
Canto - Recuerdos
63
Bar solo
ra - tion.
a a
/ɑ - u - o/
canto violin

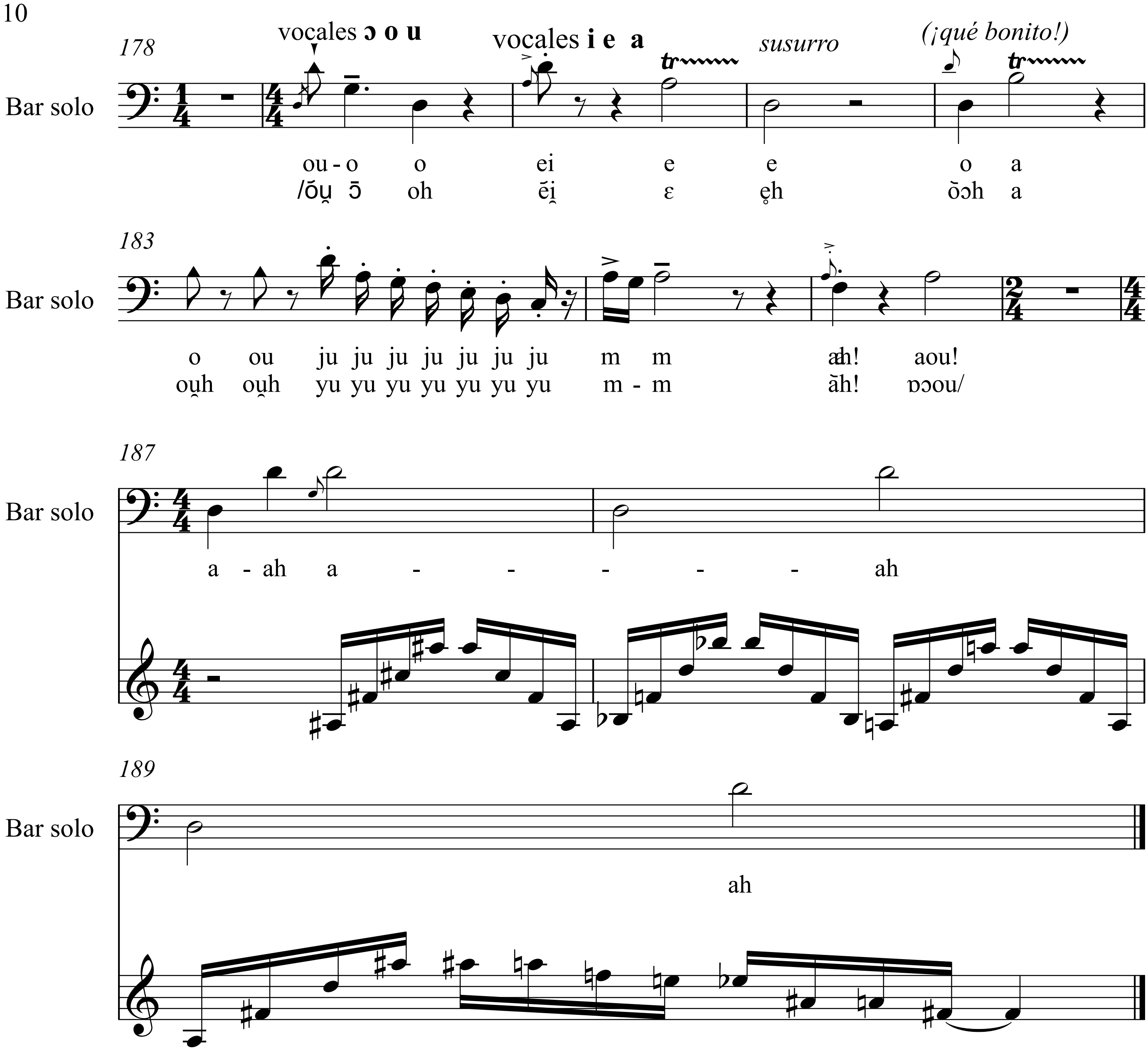

67
Bar solo
Voz
a
mp
sus vo - ces bra - man en el tiem - po
71
♩ = 100
♩ = 60
a
ruido del viento
/ħ
pp
76
♩ = 100
♩ = 70
port.
su - su - rros del vien - to.
hi̥ i - y u o u/
ruido del viento
/s
pp
82
(fricativas del Ubykh)
(Albanés: 'por qué?')
f - ɬ s ʃ ʃ s pse/
3
3


87
Bar solo
En mi me - mo - ria de la nie ve
1: Llamada - Evocación
92
Bar solo
Kroy - kha-sis Kroy - kha-sis laes-pi - ra - al
(Escita: 'Cáucaso')
96
Bar solo
del bra - mi - do del tiem - em - po
3
La duración
98
Bar solo
die - se Da-uer was war sie?
102
Bar solo
was - war sie war sie ein Ze - it- raum?


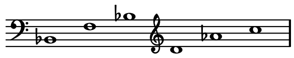
106
Bar solo
Et-was Mess - ba - res ei-ne Ge-wiss- heit?
111
Bar solo
die Da-uer was ei-n Ge fühl das
Voz
Nein!
116
4 Sinera
Bar solo
Le - bens-ge - fühl
121
Bar solo
a - a la ne-gra bar - ca
/a ε - e/


126
Bar solo
por mi vi - gi - la a - - a ve pel meu
6
3
130
vibrato
Bar solo
som - ni del mar de Si - ne - ra
3
134
5
Recuerdos
Bar solo
als meus ulls
17
137
Bar solo
ja no sa- ben sa - ben
9
7
140
Bar solo
si - nó con - tem - plar - r di - es i sols per -


144
(elegir la vocal más resonante según el registro)
Bar solo
duts
a - - - - a
/ɑ - ɔ o ɑ - ɑ/
147
Bar solo
com sen - to ro - dar
9
150
Bar solo
ve - lles tar - ta - nes pels ri - als de Si - ne -
5
3
Sinera : Areyns
/i -ɛ- ə : ə - ɛ̃ŋs/
6
155
Bar solo
ra
Si - ne - ra Si - ne - ra Si -
/i - ɛe - ə/ i - ɛː - ə i -
6
6


159
Bar solo
ne - ra
Si - nera
Si - ne - ra
Si - ne - ra
εeø - ə
i - eøə
i - εø - ə
i - ε - ɵ
163
(Ubykh)
(tiempo
para respirar)
Voz
saa
swa - swa
ʃʷa
ʃʷa - ʃʷa
('mar')
('Mediterraneo)
Solo se pronuncian las vocales
167
Bar solo
A - reyns
A - reins
A - reyns
ɒ - ɑ
ɑ - a
ɑ a aɛ
171
molto
expressivo
Bar solo
A
reyns
a - œ - a
molto expressivo
174


178
Bar solo
vocales ɔ o u
vocales i e a
susurro
(¡qué bonito!)
ou - o o ei e e o a
/ō̃u̯ ɔ̄ oh ẽ̆i̯ ɛ e̥h õɔh a
183
Bar solo
o ou ju ju ju ju ju ju ju m m aħ! aou!
ou̯h ou̯h yu yu yu yu yu yu yu m - m ăh! ɒɔou/
187
Bar solo
a - ah a - - - - - ah
189
Bar solo
ah

Voice prints (VP) is a homage to our ancestors and the languages they used as tools to create and expand over large locations powerful civilisations. Ubi sunt? Where are our ancestors now? Have they disappeared? Those people and the locations they lived in are casted in infinite events created by our thoughts. We can hear the resonances of their voices in the roar of time. In VP the ancestors' voices are articulated by the voice of a baritone and of a violin who melt their timbre in resonances of the words uttered by a distant father: harmonics 2, 3, 4, 5, 7, and 9 of a fundamental Bb$_1$ tone:

How do we get to know them? We get acquainted with them by picking up out of those resonances the identifying features from which the voices of ancestors are assembled in our minds. We invite you to listen what they have to say to you and hence bring them back from an alleged Atlantis. VP includes phonemes, words and sentences sung and spoken in Scythian, Greek, Celtiberian (Luján, 2006; Rodríguez Ramos, 1997; Villar, 1995), Ubykh (Charachidze, 1989; Dumezil, 1931; Fell, 2012; Ladefoged, 2005), Albanian, Catalan, English, Galician, German, Spanish. Music is set to fragments of poems by Espriu (1946) (*Cementiri de Sinera, poems 4, 17*), Handke (1986) (*Gedicht an die Dauer*) and Llamazares (1982) (*Memoria de la nieve, poem 2*).

Text in the composition to be sung or spoken

Section 1: Questions, invocations, calls:
Qu-word: *Ubi* (Lat), *où* (Fr), *onde* (Gal), *dónde* (Sp), *wo* (Ger), *where* (Eng), *ku* (Albanian), *opoú* (Class. Gr), /maːk'ʲa/ (Ubykh)[1]

(1) a. *¿Dónde están?*
(Spanish)

b. *Ubi sunt?*
(Latin)

c. *On són?*
(Catalan)

d. /mạ-ø-χa-na-y/
where-3PA-be.PL-PL-Q
'Where are they?'
(Ubykh)

e. *Ku janë ata?*
(Albanian)

f. /sa.ø.ʃ'.q'á.y/
what-3SA-become-PRT-Q
'What happend?'
(Ubykh)

Section 1: Ancestors:

(2) Celtiberian names (García Quintela, 2005)

a. Eburia[2] Kalueni

b. Apanus[3] Amboli

c. Api[4]

(3) Celtic names

a. Fénius Farsaid (king of Scythia)

b. Goidel Glas

c. Sru

d. Brath

e. Breogan

[1] Ubykh at UCLA: recording of speaker Tevfik Esenç made in 1986 by J. C. Catford.

[2] Celtic female name used in Galicia. Derived from "-ya", "eburo", a celtic word refering to the coniferous tree 'taxus baccata', 'tejo común ibérico' (texeiro, gal.). /beriya/, iberia

[3] Brother of Apan. Apana is a female name derived from Api.

[4] 'mother', 'water'. Name of Scythian goddess.

Section 2: Chant. Events of memory evoked through language

Language (Eng), Glossa (Greek), /bʰzḁ/ (Ubykh), Sprache (German), Gjuhë (Albanian).
They are chained in an event of infinite duration.
Sus voces braman en el tiempo
Susurros del viento
/hi̥ i-y u o u/
/s f ɬ s ʃ s/ (Ubykh fricative consonants)
pse (Albanian: 'why?')
En mi memoria de la nieve.
Kroykhasis (Scythian)
En la espiral del bramido del tiempo. (Julio Llamazares, Memoria de la nieve)

Locations where memory dwells

(4) Section 6: Areyns[5]

a. Sinera
/si.nέ.ɾə/

b. Areyns
/ə.ɾέɲz/

c. /ʃʷa/
'mar' (Ubykh)

d. /ʃʷaʃʷá/
Mediterranean Sea (Ubykh)

(5) Scythia

a. Kroy-khasis
'Caucasus'[6]

(6) Artabrian Coast

a. Arrotrebae
/ar.trəβ.œ/
'Artabria'

b. Artabri
'the artabrian people'

c. Brigantis
/bɾi.ɣ̞ãn.tis/
'Brigantia'[7]

d. Ωκεανος
okeanós
'Ocean'

e. Atlantis[8]

[5] Sinera, anagram of Areyns de Mar.

[6] Pliny the Elder in *Natural History (77-79 A.D.) attributes a Scythian origin to the name Caucasus "kroy-khasis" meaning '(mountain) ice-shining with white snow'. Online Etymology Dictionary:* http://www.etymonline.com/index.php?term=Caucasus

[7] Name given by Breogan to the city of A Coruna.

[8] Mythical island described in Plato's dialogues Timaeus and Critias, which allegedly became submerge into the Atlantic Ocean.

References

Charachidze, Georges. 1989. Ubykh. *Pages 357 – 459 of:* Hewitt, B. George (ed), *The Indigenous Languages of the Caucasus. Vol. 2: The North West Caucasian Languages*. Delmar, N. York: Caravan Books.

Dumezil, Georges. 1931. *Le langue des oubykhs*. Paris: Champion.

Espriu, Salvador. 1946. *Cementiri de Sinera*. digital book edn. Barcelona: La Sirena.

Fell, Brian. 2012. Case and agreement in Ubykh. *ms*.

García Quintela, Marco V. 2005. Celtic Elements in Northwestern Spain in Pre-Roman times. *e-Keltoi*, **6**, 497–569.

Handke, Peter. 1986. *Gedicht an die Dauer*. Frankfurt am Main: Suhrkamp.

Ladefoged, Peter. 2005. Features and parameters for different purposes. *UCLA Working Papers in Phonetics*, 1 – 13.

Llamazares, Julio. 1982. *Memoria de la nieve*. Consejo General de Castilla y León.

Luján, Eugenio. 2006. The Language(s) of the Callaeci. *e-Keltoi: Journal of Interdisciplinary Celtic Studies*, **6**, 1–34.

Rodríguez Ramos, Jesús. 1997. Sobre el origen de la escritura celtibérica. *Kalathos: Revista del seminario de arqueología y etnología turolense*, 189–198.

Villar, Francisco. 1995. *Estudios de celtibérico y de toponimia prerromana*. Vol. 260. Universidad de Salamanca.

Des pas sur l’invisible

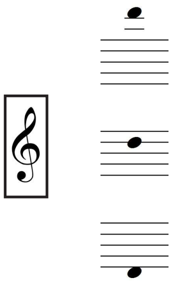
for Clarinet or Saxophone

Sílvia Teles

Work created in the context of the Frederico de Freitas
Interpretation Prize (2016), Aveiro University.

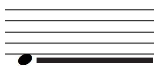
Cover painting: Sofia Gomes. Solaris I, 2016.

Notation



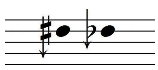
The score comprises three staves corresponding to three main polarizing sounds: D*4*; D*5* and D*6* (transposed sounds) the fundamental, 2nd harmonic and 4th harmonic respectively.
The importance of the octave interval between these three sounds is thus established, on the one hand, as the limit of a space to be filled or traveled, and on the other hand as an intensive space: where different tensions, movements and temporalities are generated.

In the space thus defined the following main effects are attributed:

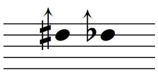


Continuously sustained note without vibrato (S.V.).

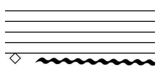
Momentary interruption of the actual presence of the note D*4* (transposed sound).



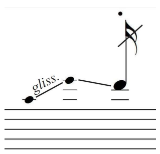
Quarter tone below.



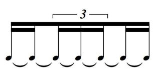
Quarter tone above.



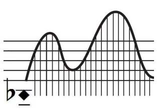
Sung sound in glissando between the annotated notes. It is also transposed. If necessary, sing to the octave.



Glissando. Do not emphasize intermediate notes.



Measured vibratto.



Air sounds.

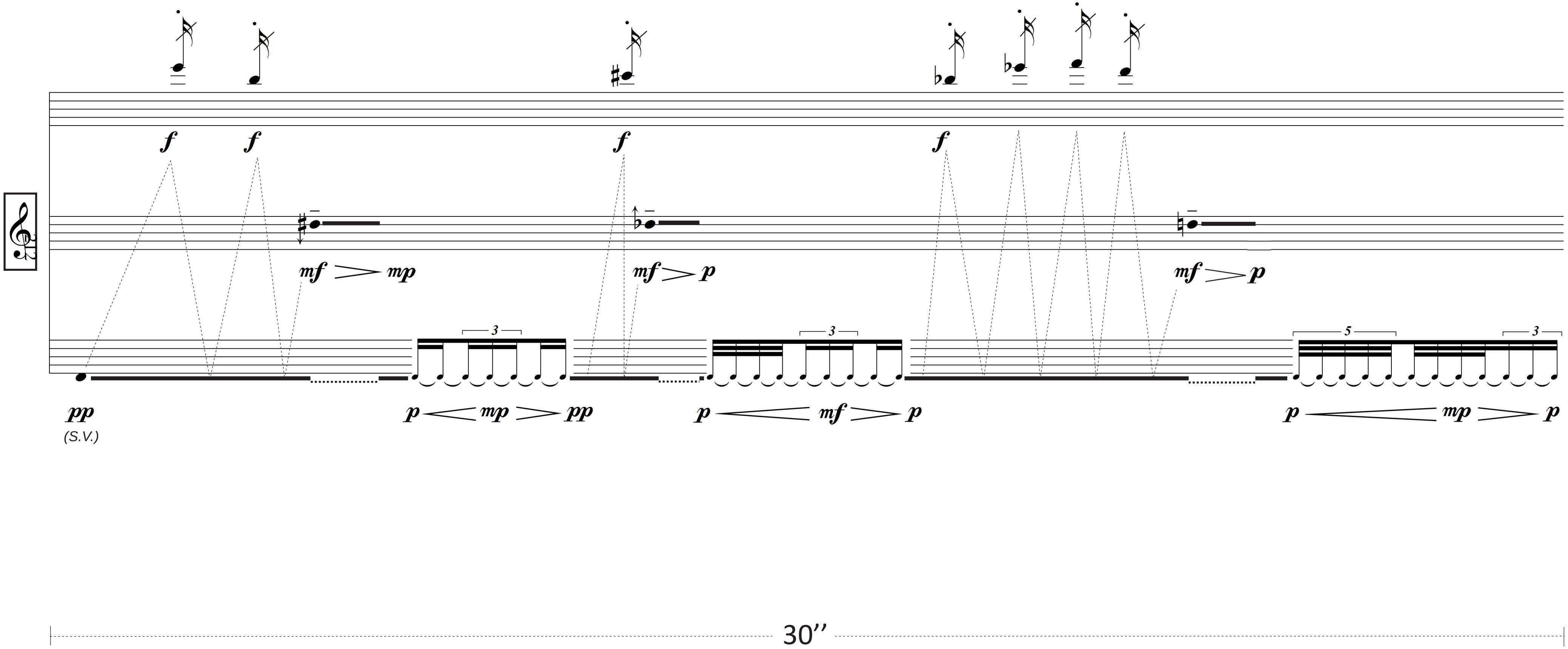
Duration: 4' cca.

Des pas sur l'invisible

for Clarinet or Saxophone
2016

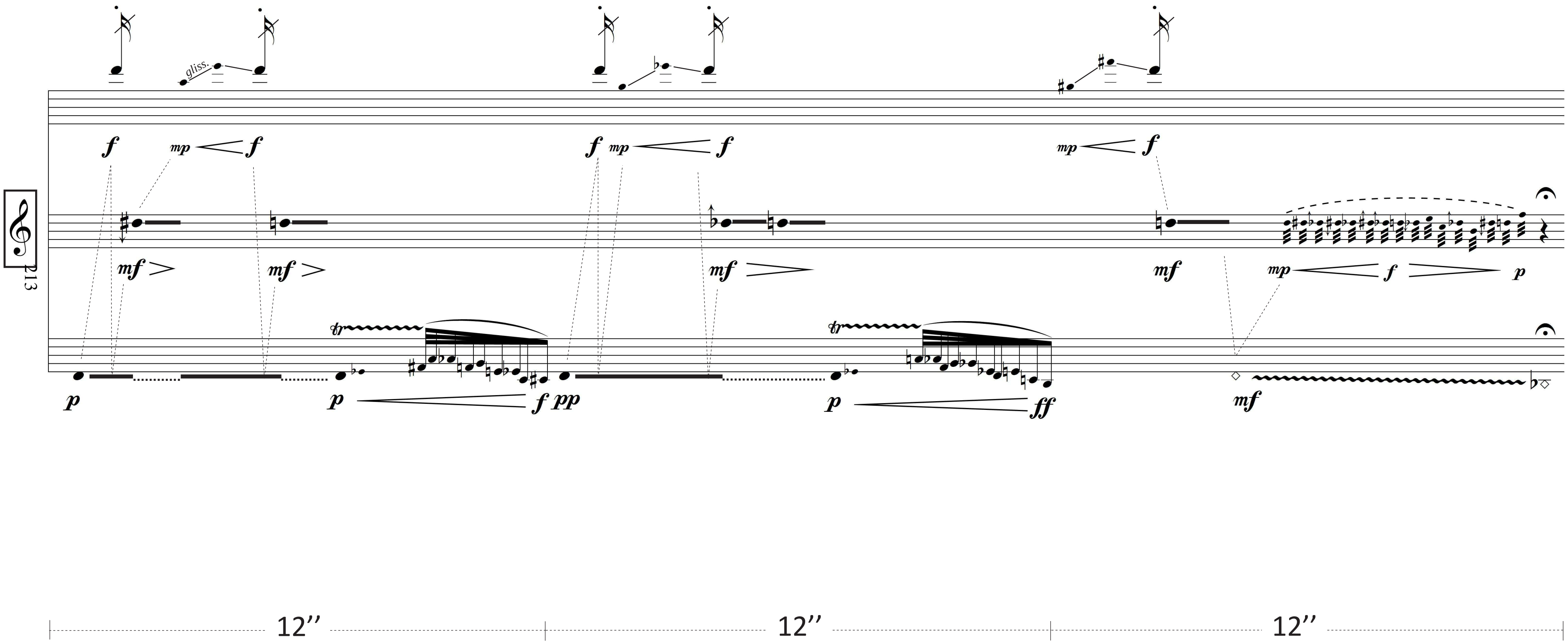
Sílvia T

"Seen from the timelessness of the dream, time is aperture, way of acess and way to walk.[1]

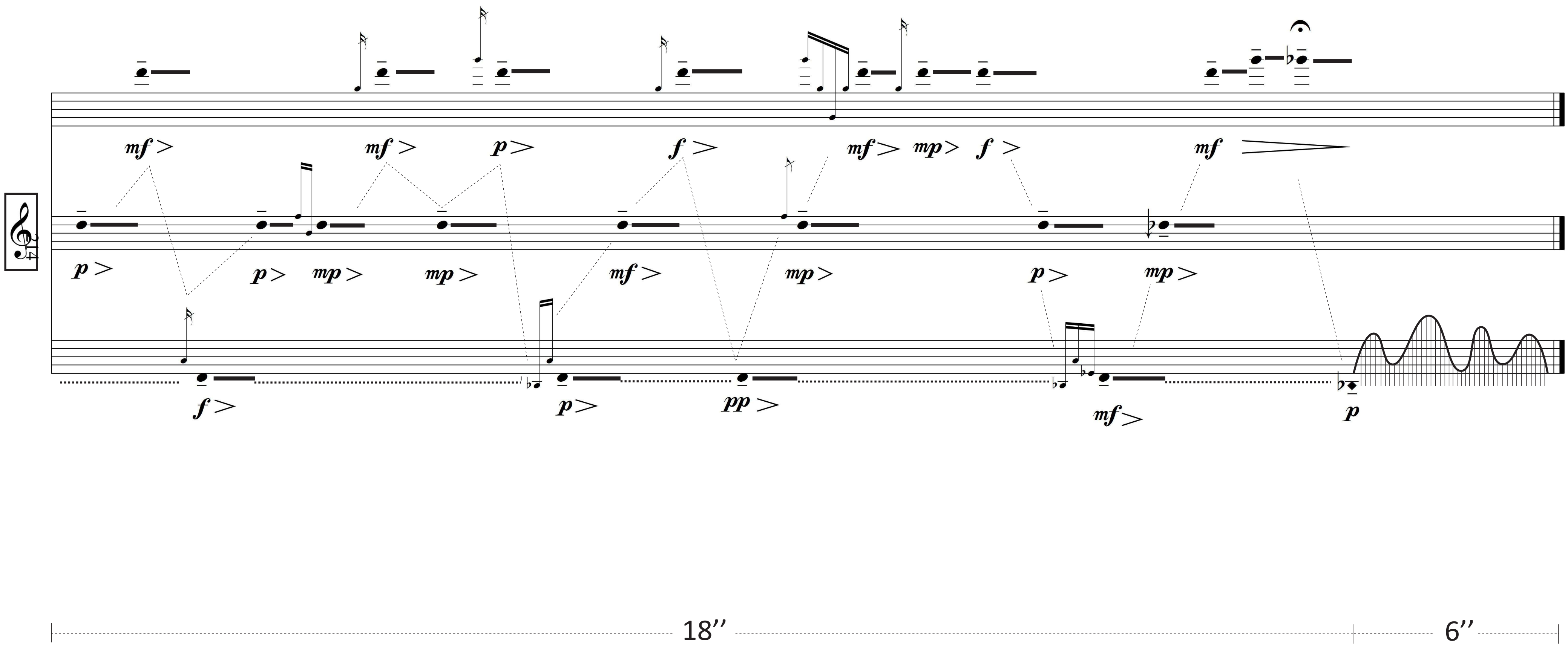


* *Transposed sounds.*

[1] ZAMBRANO, Maria. 1994. Os sonhos e o tempo. Relógio d'Água. Lisboa.



213
gliss.
f
mp
f
mf
p
pp
ff
12''
12''
12''


mf
p
mp
f
pp
18''
6''

TERRAFORMATION: FOR VIOLIN OR VIOLA AND COMPUTER

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

ABSTRACT

This paper introduces my real-time notation (RTN) work *Terraformation* (2016–17) for violin or viola and computer. Program notes, performance directions, and two score excerpts from violinist Florian Vlashi's performance on May 25, 2017 at the Third International Conference on Technologies for Music Notation and Representation are included.

1. PROGRAM NOTE

Terraformation concerns a fusion of several disparate themes. The first, and perhaps central, theme is that of terraforming. This is the hypothesized large-scale transformation of an inhospitable planetary body into one fit for Earth-like organic life. Popularized in science fiction, serious studies on the procedures for terraforming come from the gradually maturing scientific exploration programs on Earths moon, Mars, and Venus. These issues prompt reflection on humanity's history of colonialism, abuse of resources, lack of environmental concern, and how these might manifest beyond our home planet.

At the same time, *Terraformation* is inspired by Philip Johnsons sculptures and architecture at the Fort Worth Water Gardens in Fort Worth, Texas. This urban park contains several named "micro-environments": Active Water Pool, Aerated Water Pool, Quiet Water Pool, Mountain, Central Square, Stage, and Events Plaza. The style of the Gardens is minimal and angular. They give an abstracted impression of a natural landmark such as a mountain or a river canyon, ignoring many realistic details in favor of sensory appeal.

The connection between terraforming and the Fort Worth Water Gardens is humanity's attempt to fashion a world after its own design. This world has rough edges and missing details, no oceans and preciously little oxygen. Everything is synthetically derived. We bring our plants and animals, our histories and cultures. We also bring our diseases, our selfishness, and our unchecked ambitions. *Terraformation* is a creation story.

This piece uses a computer screen to display music notation that changes during the performance based on decisions made by both the musician and the computer. In this way, every performance is unique and unrepeatable.

2. PERFORMANCE DIRECTIONS

2.1 Performance Overview

Terraformation uses real-time notation and requires the performer to sight-read music as it is algorithmically generated during the performance. The goal of a performance of the piece, therefore, is not perfect adherence to the demands of the score, but a productive interaction between human and artificial intelligence. The performer should attempt to both read the music as accurately as possible and respond to it intuitively, which will in turn influence the computer's musical decisions. The piece is "cartographically" composed meaning that the large-scale structure is mapped by the composer but the surface details are left to the computer and performer to determine. The violist drives the notation forward by briefly depressing a MIDI foot switch. Pressing and holding the foot switch down will cycle through alternate paths through the piece. The pacing and direction of the piece are thus determined by the performer.

2.2 Real-Time Audio

A computer-generated audio component is generated live during the performance. A microphone placed near the performer allows the computer to analyze the performance which then influences the resulting computer-generated audio. In addition, the acoustic sound of the viola is both amplified and processed by the computer.

2.3 Real-Time Notation, Sight-Reading, and Improvisation

The notation is generated in the moment of performance and requires the performer to sight-read the notation in front of an audience. This is an incredibly vulnerable act to ask the performer to engage in. The goal of a performance of the piece, therefore, is not about perfect adherence to the demands of the score, but about the collaborative interaction between human and computer. The performer should attempt to both read the music as accurately as possible and respond to and influence the computer's musical decisions.

Although the notation for *Terraformation* is displayed with a great deal of precision, the composer realizes that the high demands of sight-reading might place the musician in a situation where a completely accurate rendering of the notation will result in a stilted performance. On the other hand, this piece requires no improvisation. For this reason, the CPN elements are supplemented with the fingerboard diagram and color gradients. It is the composer's hope that these additional notational elements can be read simultaneously so as to efficiently read the notation quicker and more accurately.

In conversations with violist Michael Capone, he narrated his music reading experience. He would often consult the fingerboard notation at the start of a new system, approximately placing his fingers while beginning to move the bow. Next, he would quickly assess the rhythmic figure and shape of the gesture, and begin playing the approximate rhythm and gesture. Finally, he would closely read the CPN, refining his hand position, rhythm, gesture, and other playing parameters in the process. The entire procedure could be summarized as approximation moving toward accuracy over the course of each new system of notation.

2.4 Reading Notation From A Display

Due to the real-time nature of the notation, the musical directions must be read from a computer display. In order to facilitate ease of use for the performer, the software that must be run during the performance is divided into two applications: the score application, where real-time notation will appear for the musician to read and perform, and the audio application, where a microphone input, speaker outputs, and computer processing levels are set.

2.5 Rehearsal and Example Scores

Despite the fact that you will be sight-reading during performance, this piece requires rehearsal. Rehearsal with the software will give the performer a general sense of how the piece unfolds, what you might expect to play, and an ear for the types of interactions available between computer and performer.

If rehearsal with the software is not possible, the composer can provide several example scores. These are intended to provide the performer with a sense of the work and not to be used as live performance scores.

3. PERFORMANCE HISTORY

Terraformation was premiered by violist Michael Capone on April 24, 2017 at the University of North Texas. A video of the premiere performance is available here: https://youtu.be/wrAcQiGzvVQ.

Florian Vlashi preformed the premiere of the violin version of *Terraformation* at the Third International Conference on Technologies for Music Notation and Representation on May 25, 2017 at the University of A Coruña, Spain.

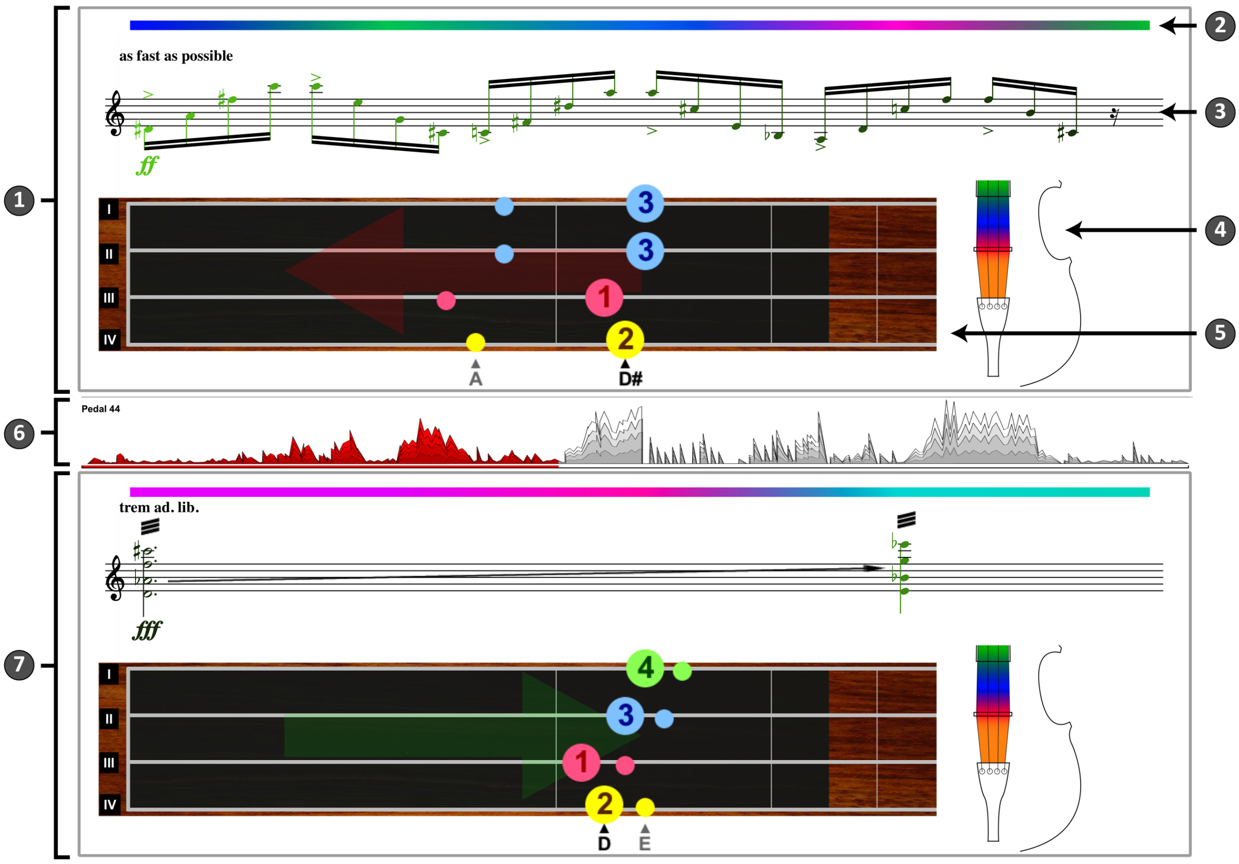
4. ACKNOWLEDGEMENTS

The composer would like to thank violist Michael Capone for his enthusiastic collaboration in the development of this piece.

5. FURTHER QUESTIONS

Please address all further questions and concerns directly to the composer at sethshafer@gmail.com. Please contact directly for links to download the performance software and example scores.

Score – Notation Window



1. Current Staff System: The current location in the piece is displayed in the upper section of the GUI.

2. Bow Contact Position Gradient: This color graphic informs the player where to place the bow on the instrument. The color matches a location shown on the viola graphic (4) and should be read left-to-right in vertical alignment with the common practice notation (3).

3. Common Practice Notation (CPN): The traditional symbols for pitches, rhythms, articulations, dynamics, and other playing techniques are displayed here. Text indications for tempo and character are notated in the top left corner.

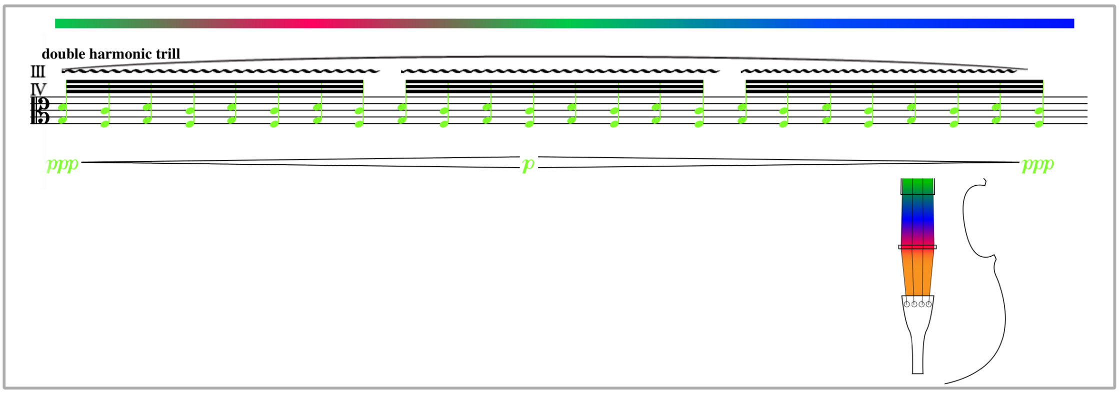
4. Viola Graphic: This graphic serves as a reference for the bow contact position gradient (2) showing the physical locations of the different colors.

5. Fingerboard Notation: This is a pictorial representation of the viola's fingerboard. Each finger is notated with a corresponding number and color on each of the instrument's strings. The pitch of the lowest string is displayed in black below. Further, the player may be asked to slide the hand position along the fingerboard to an ending location indicated by small, colored circles. The pitch of the lowest string at the terminus of a glissando is displayed in grey below.

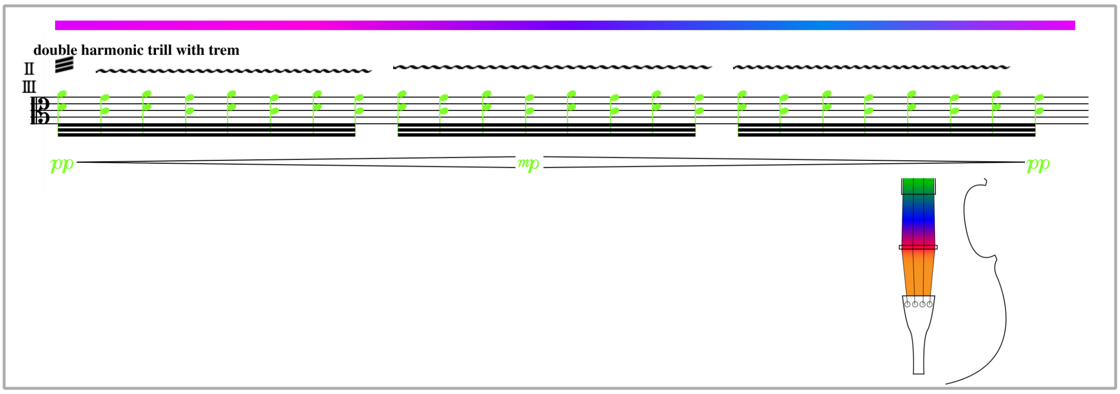
6. Formal Map: This graphic informs the player of their current location in the overall form. The red bar progresses from left-to-right at each press of the foot switch. The vertical axis of the graphic indicates expected areas of intense rhythms, dynamics, or range. The current system number is displayed in the upper left hand corner.

7. Read-Ahead Staff System: The lower section of the GUI allows the player to read ahead and anticipate upcoming material.

Performance Techniques

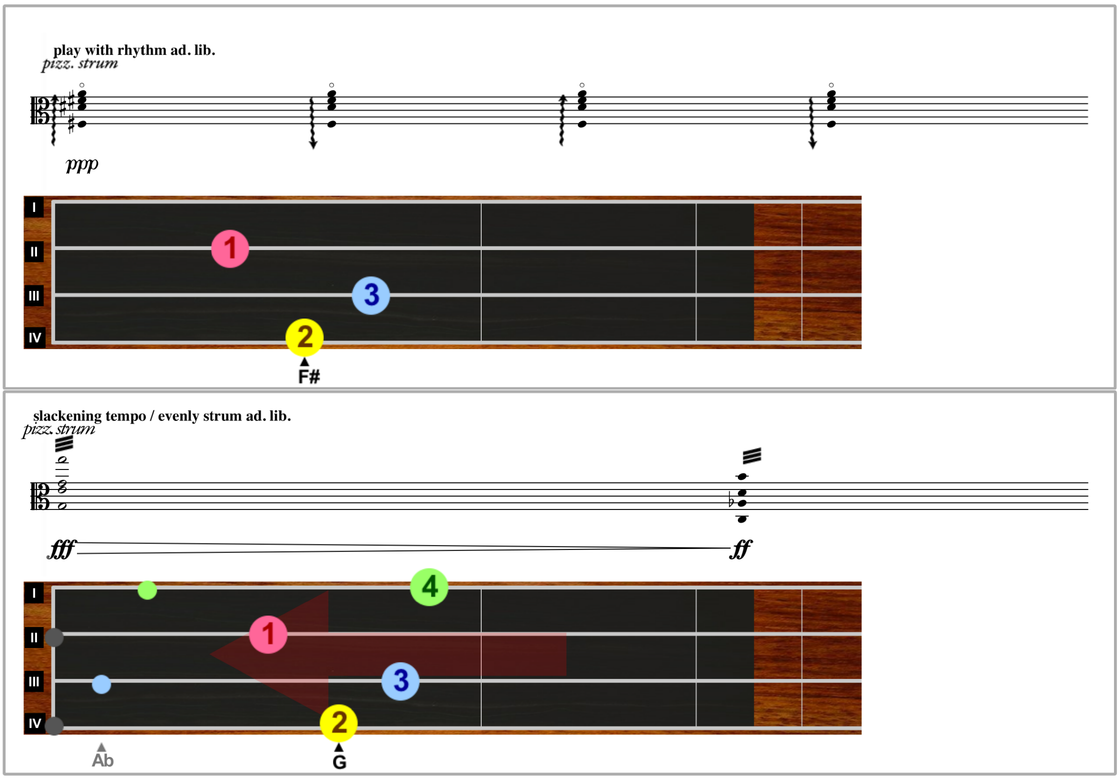


Double harmonic trill: quickly trill between two double-stopped harmonics using a legato bow. Pitches indicate fingering location. Duration of the trill can be determined by the performer rather than the exact number of notes.

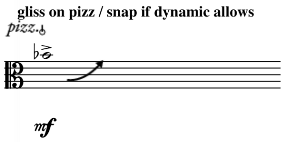


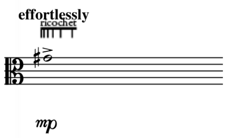
Double harmonic trill with tremolo: quickly trill between two double-stopped harmonics using a tremolo bow. Pitches indicate fingering location. Duration of the trill can be determined by the performer rather than the exact number of notes.

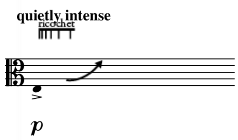
Bow behind bridge: the orange gradient above the notation indicates that the performer bow behind the bridge. Specific string and resulting pitch are indeterminate.



Pizzicato strum: pluck chord using fingers. Direction of the strum is either indicated with up or down arrows, or (as pictured here) is of indeterminate direction. Speed and character of the strum indicated in text.



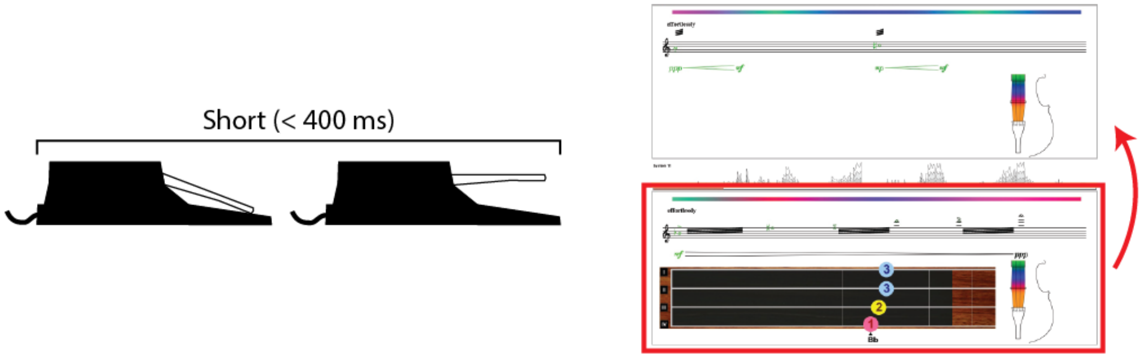
Pizzicato glissando: pluck the string and immediately slide the left hand finger in indicated direction.

Ricochet bowing: throw down bow at the string with enough force to cause the bow to bounce on the string.



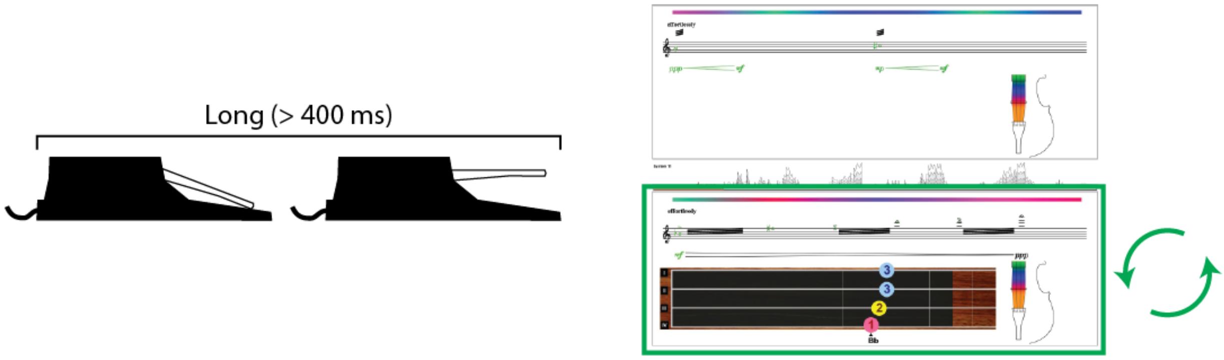
Ricochet bowing with glissando: glissando with left hand finger while performing a ricochet bow technique.

Footswitch (Pedal) Technique

The performer controls the progression of the music by depressing a MIDI footswitch. When the performer has finished playing the music on the current staff system, a quick press and release of the footswitch will cause the music in the read-ahead staff system to move up to the current staff system.



The performer can choose alternative options from the read-ahead staff system by pressing and holding the footswitch until the read-ahead staff system refreshes. The performance can execute a "long press" as many times as they want to cycle unlimited alternative music options. When an option appears that the performer would like to play, a "short press" on the footswitch will cause it to move up to the current staff system.



Flow Between Systems

In general, the performer should strive to connect each system of notation to the next to create a seamless musical experience. However, the performer is free to speed up or slow down the pace of notation advancement in any way that they deem musical.

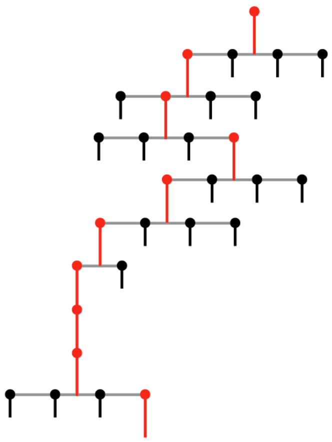
Study Scores

Overview

No single score can represent *Terraformation*. Individual performances can be captured and notated for study. Two score excerpts from the same performance on May 25, 2017 by violinist Florian Vlashi at the Third International Conference on Technologies for Music Notation and Representation are described and then presented below.

Tree Structure Score

Once selected music is selected by depressing the footswitch, the algorithms driving *Terraformation* create new notation based on the current material. The performer has the power to select what to play. This choice affects the outcome of subsequent music, which is in turn also open to performer selection. This creates a type of tree structure of performer choice where future choices are dependent on previous ones.



The notation of the piece is therefore directly shaped by the performer's selection process. The tree structure score shows the performer's choices in dark black notation connected by arrows. The light grey notation is indicative of other choices that may have been available given the number of notation-generating parameters. These parameters are printed above each system and describe the degree of variability at each moment in the piece. This single page of *Terraformation* corresponds to the first six pages of the performer's view score.

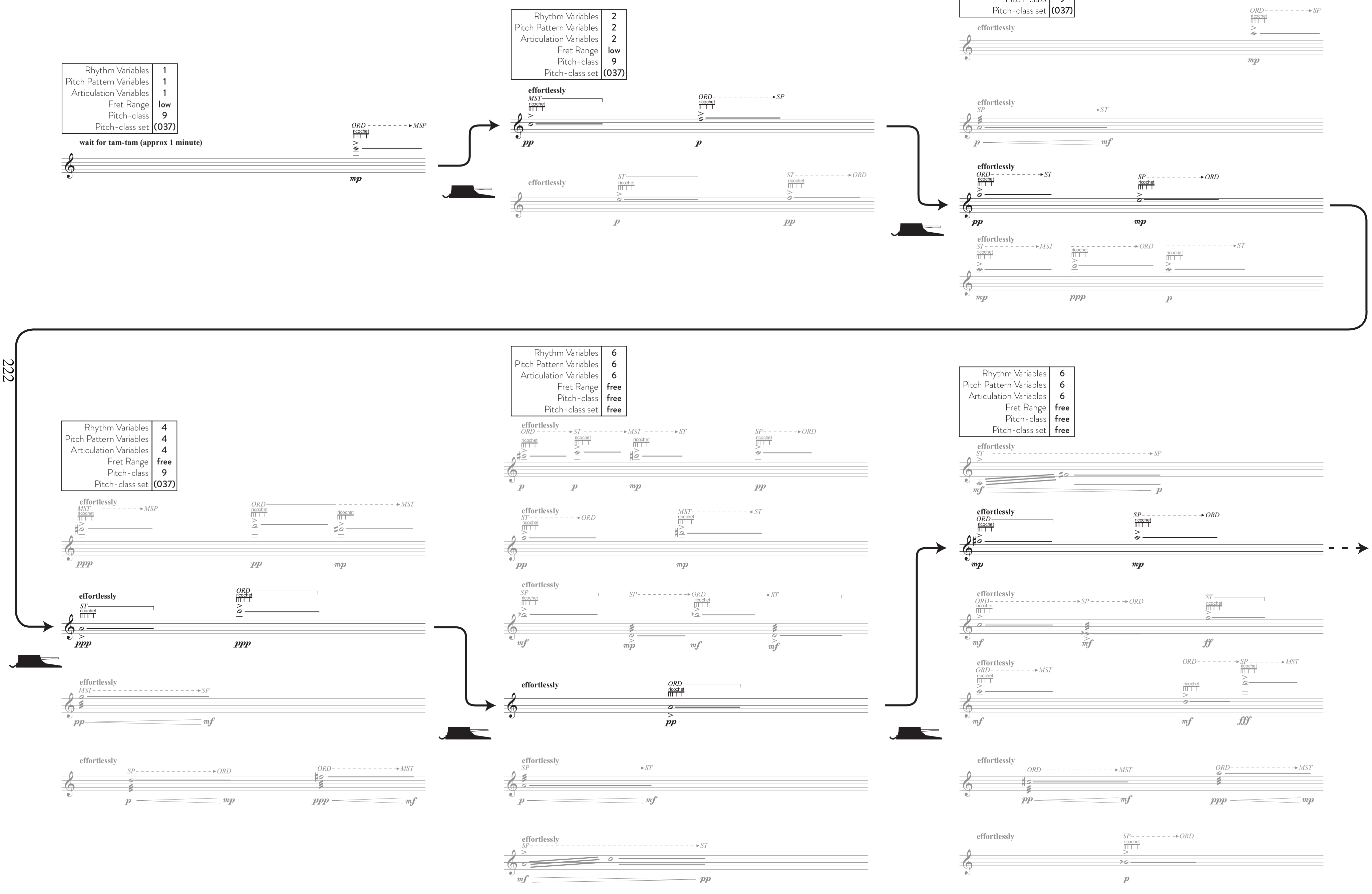
Performer's View Score

Another way to examine *Terraformation* is from the vantage point of the performer. This score captures exactly what was displayed during performance. As already described, the top system is the current staff system and the bottom system is the read-ahead staff system.

TERRAFORMATION

Tree structure score generated on May 24, 2017 at the Third International Conference on Technologies for Music Notation and Representation in performance by violinist Florian Vlashi

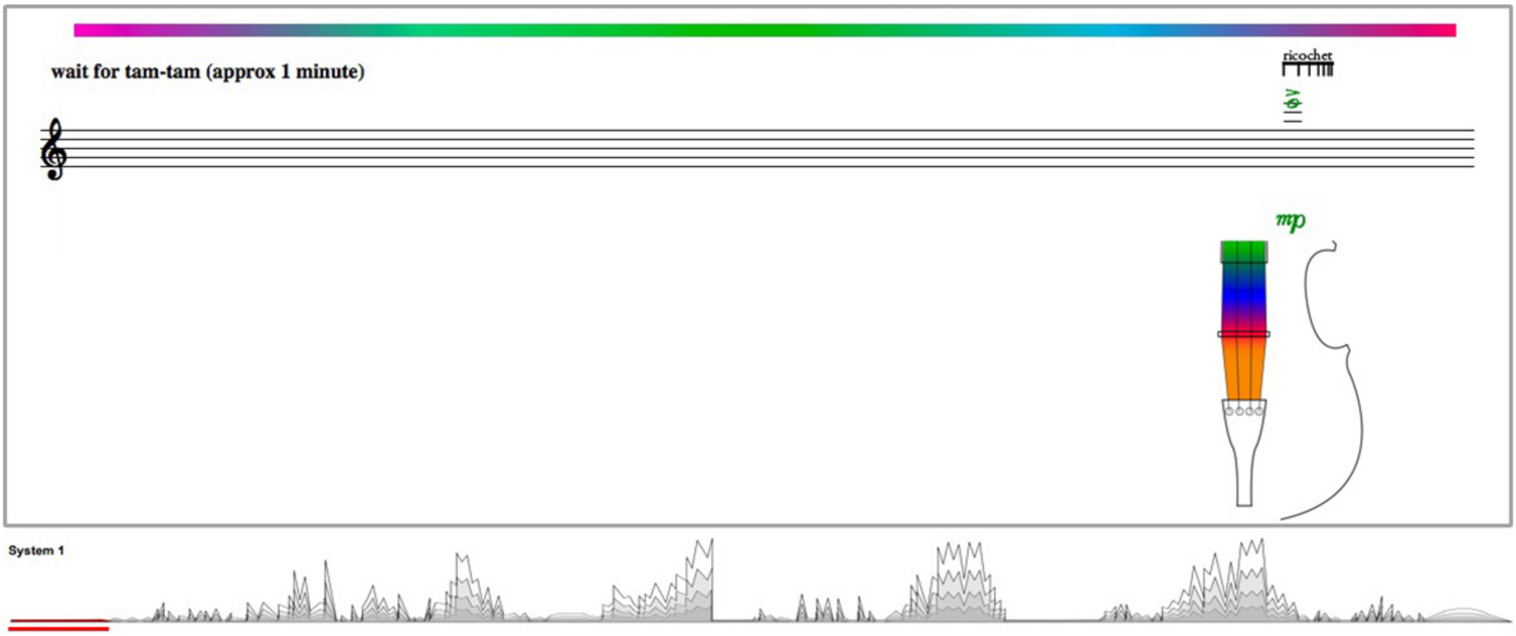
Seth Shafer

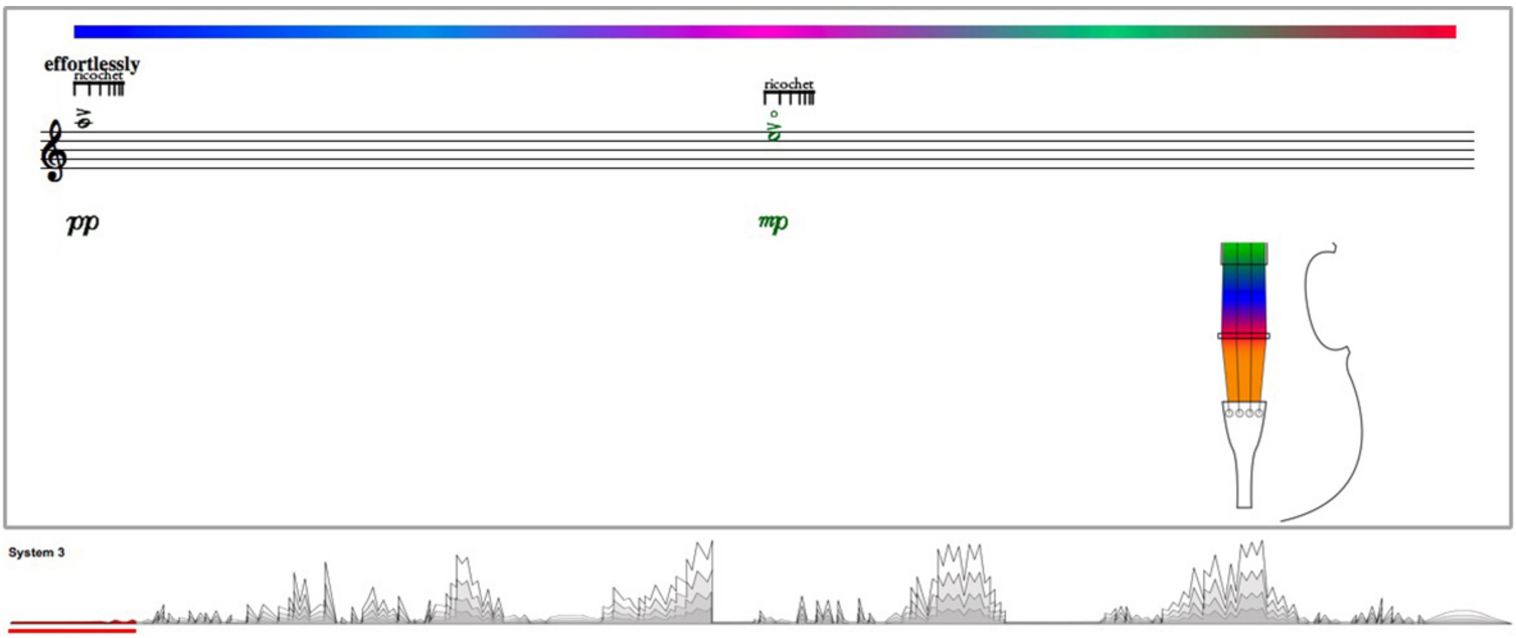


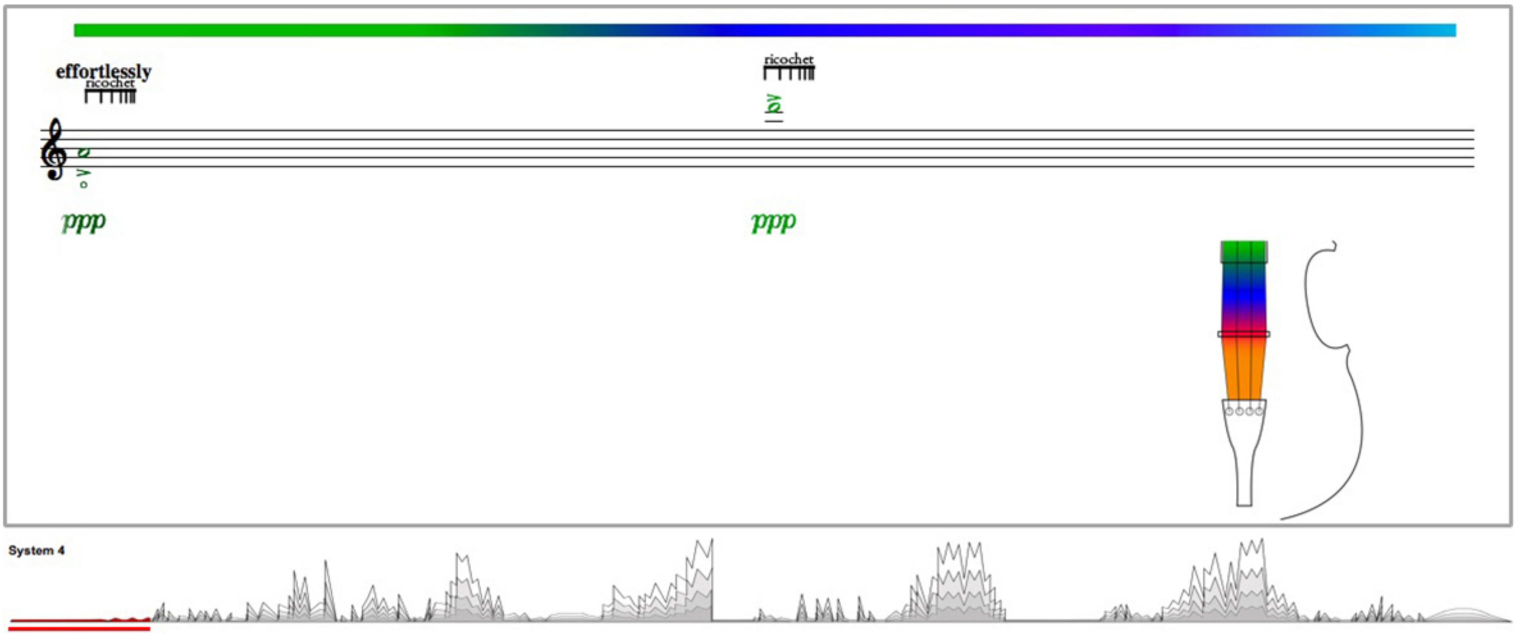
TERRAFORMATION

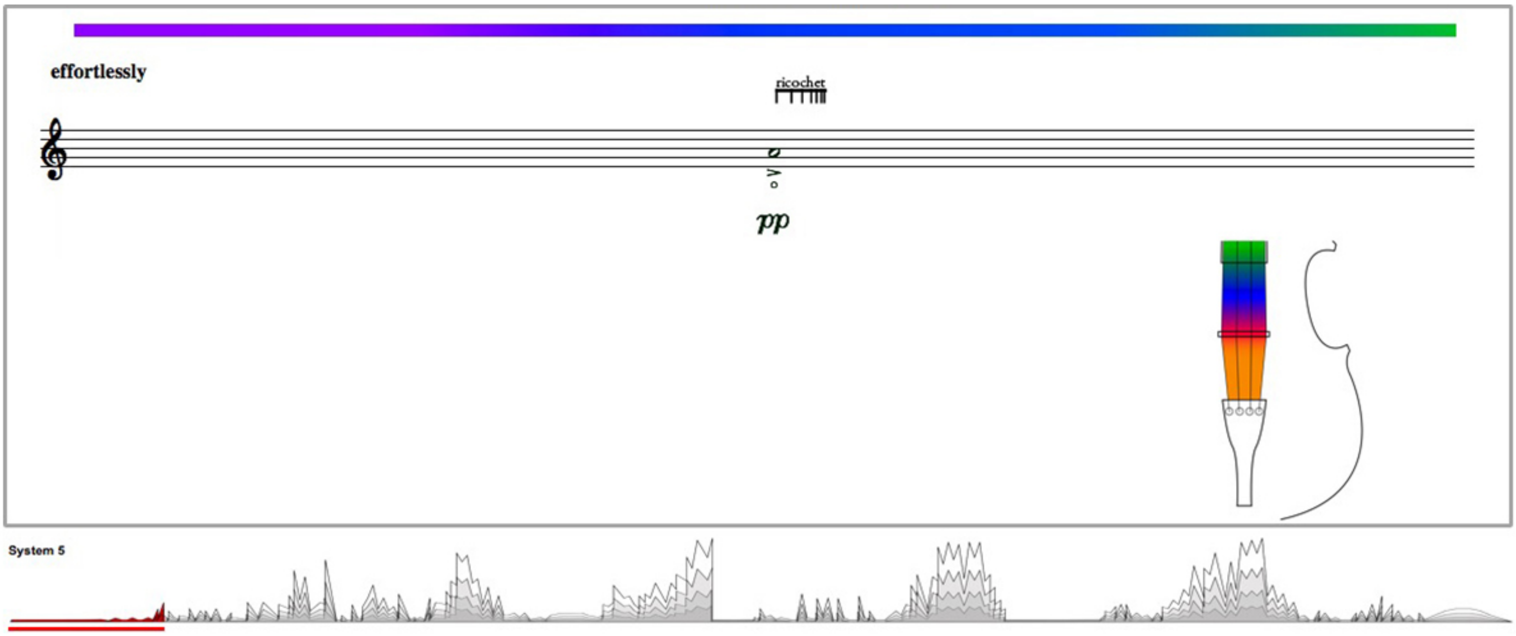
Performer's view score generated on May 25, 2017 at the Third International Conference on Technologies for Music Notation and Representation in performance by violinist Florian Vlashi

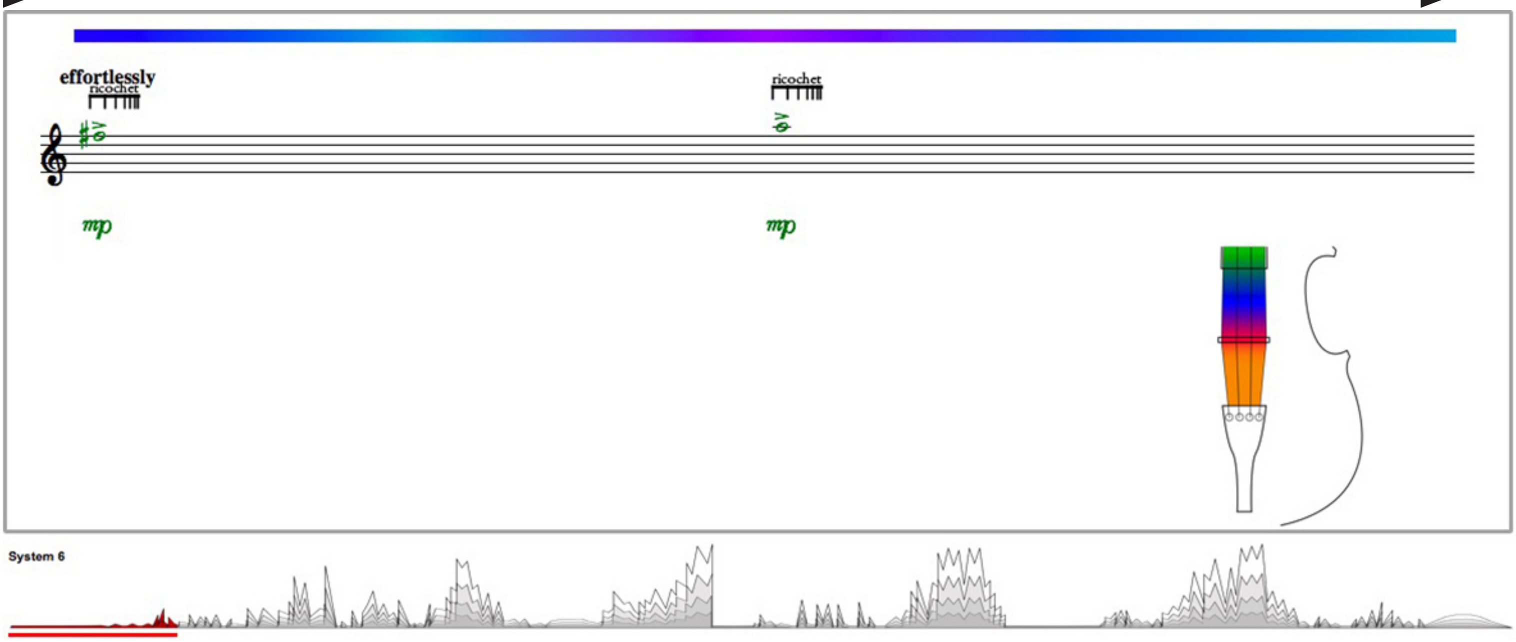
Seth Shafer

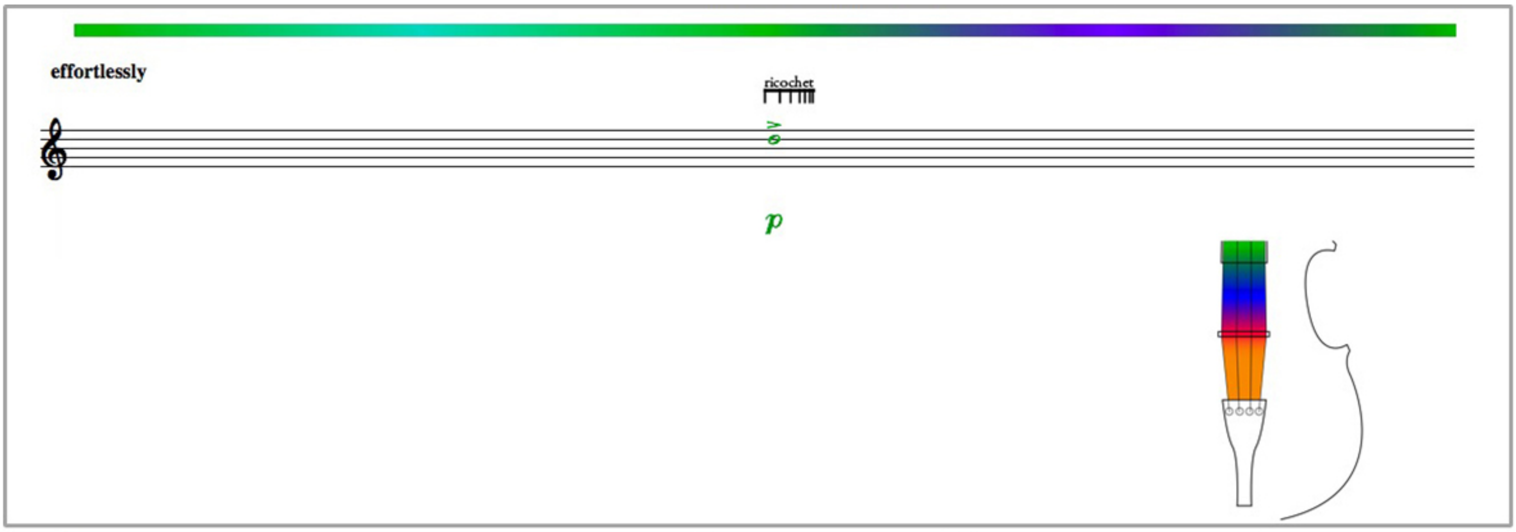




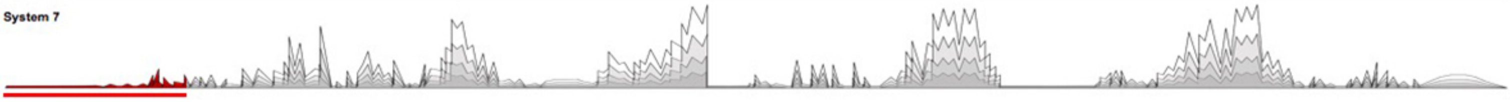

effortlessly
ricochet
ppp
ricochet
ppp
System 4

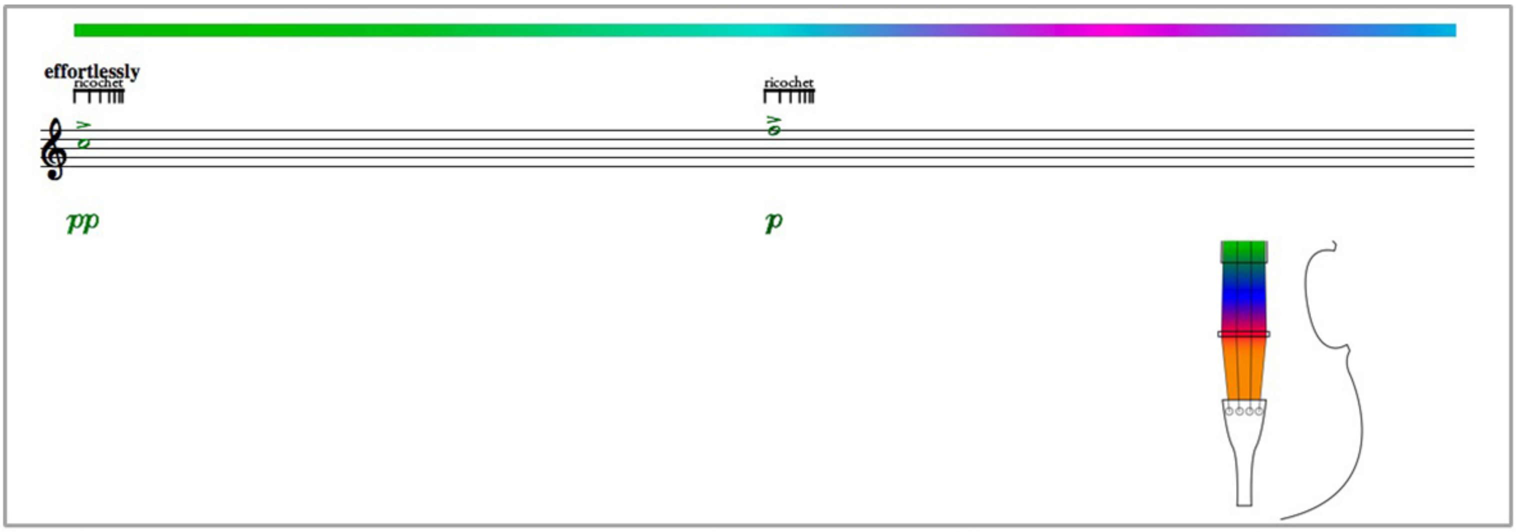

effortlessly
ricochet
pp
System 5


effortlessly
ricochet
mp
ricochet
mp
System 6

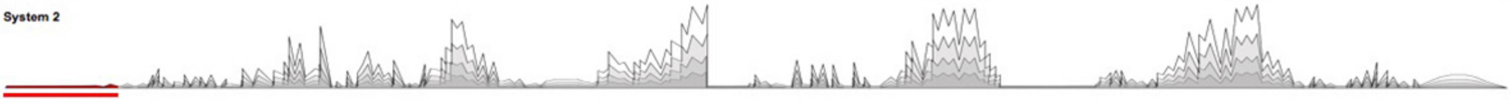


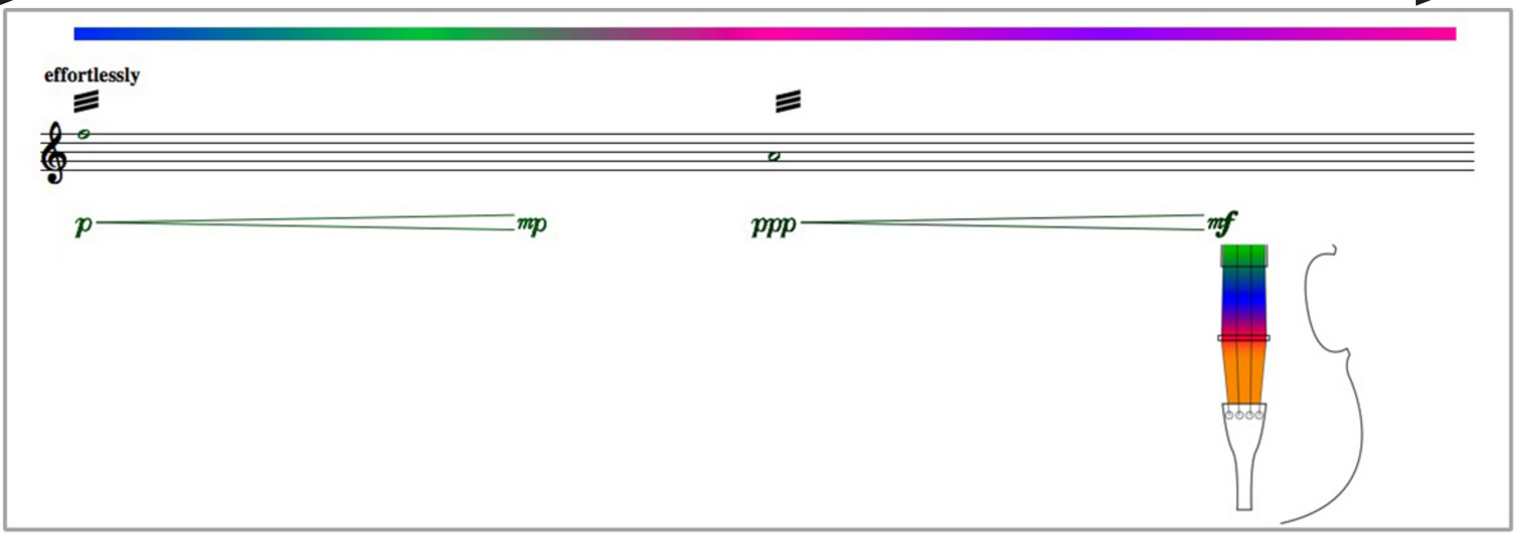
System 7



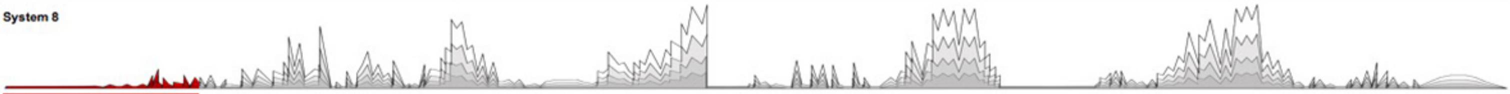


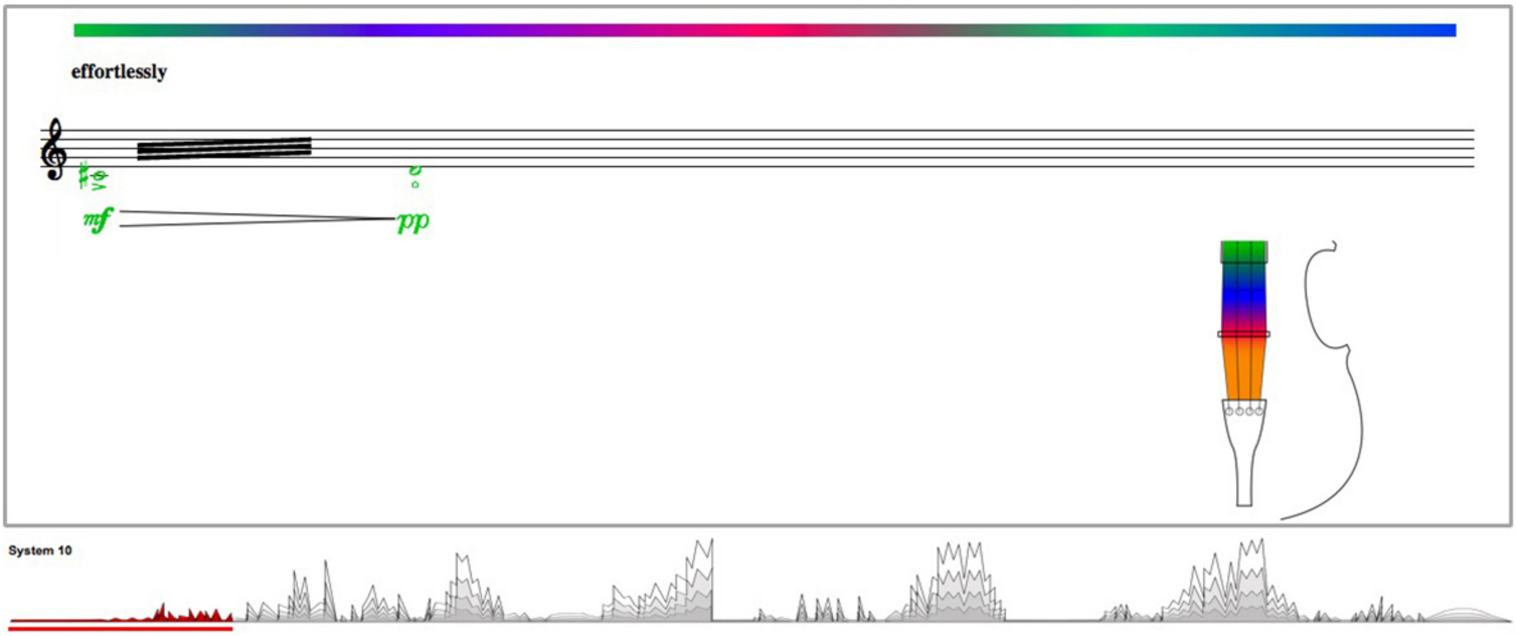
System 2

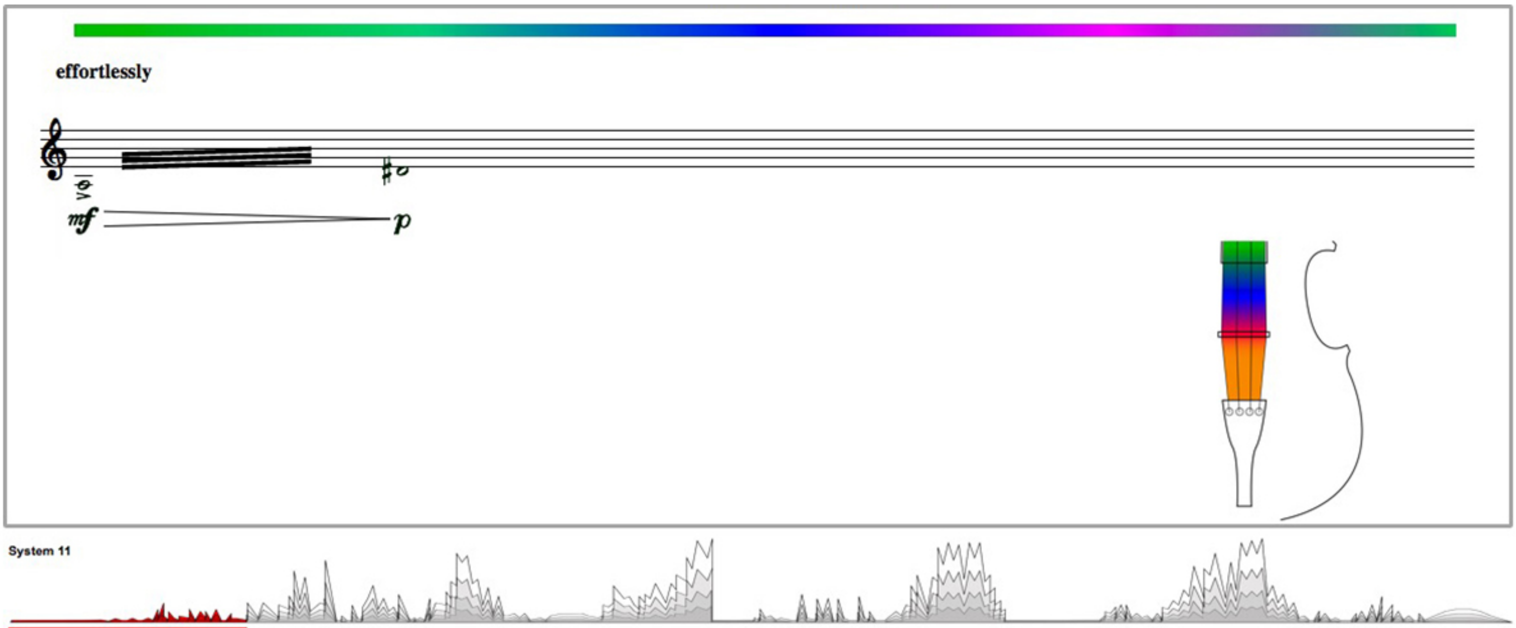


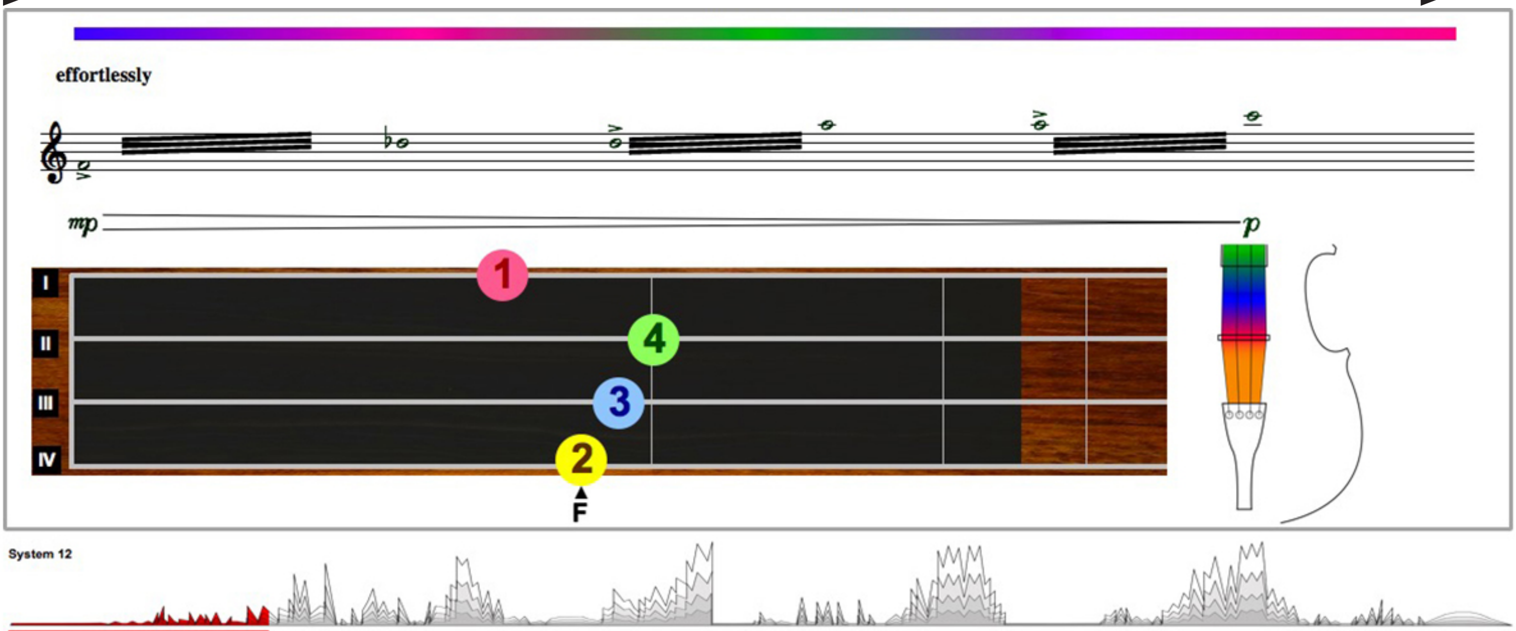


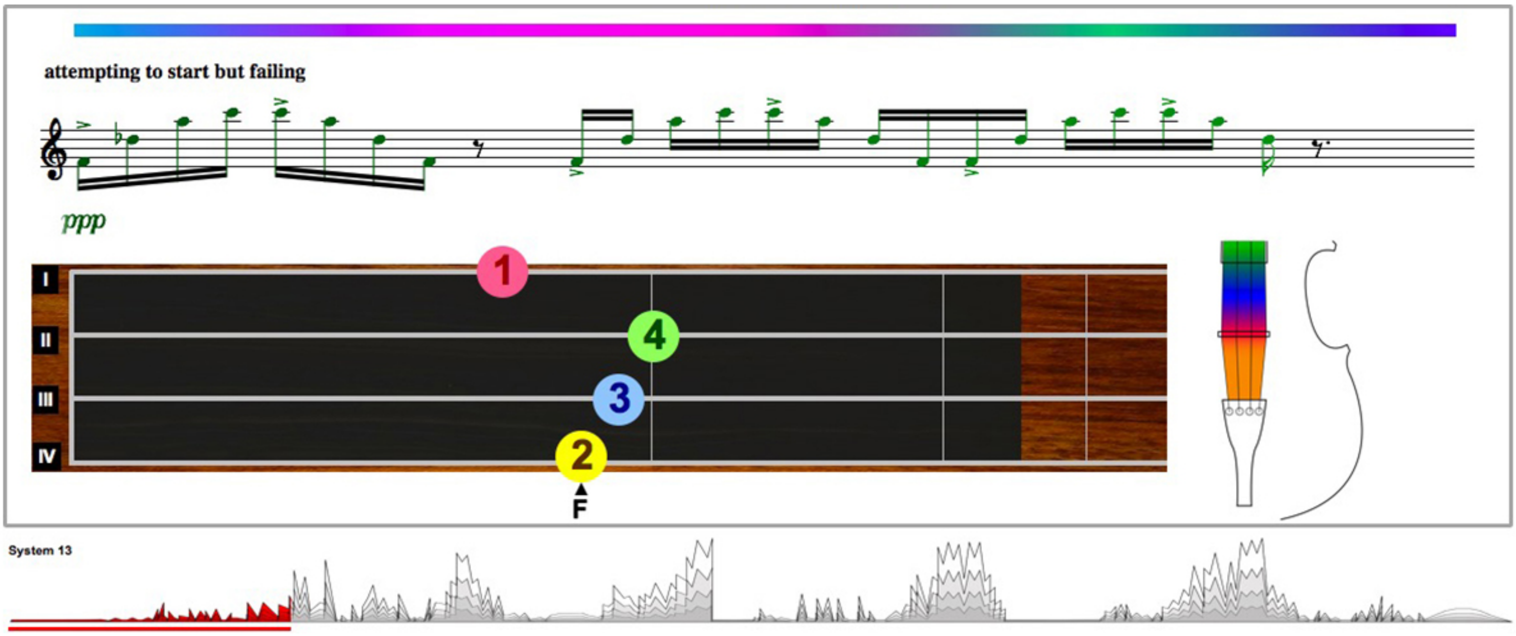
System 8

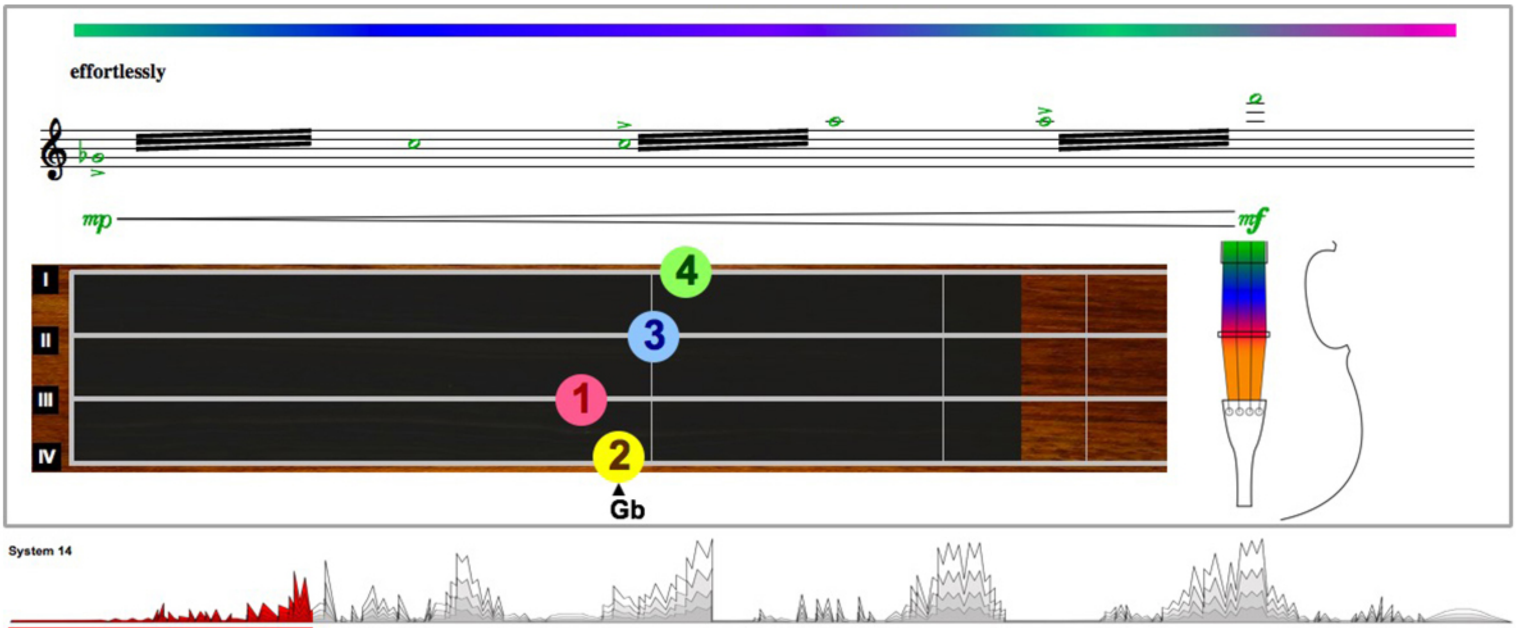


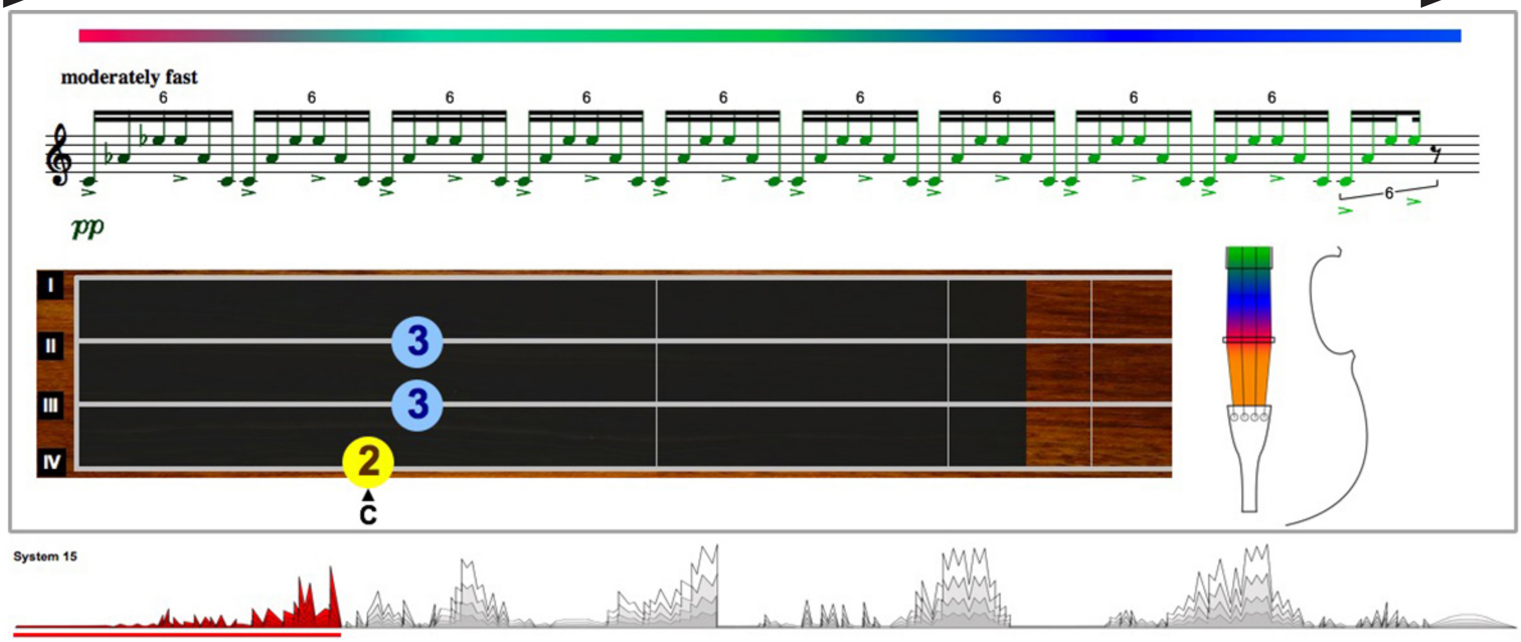

effortlessly
mf
pp
System 10


effortlessly
mf
p
System 11


effortlessly
mp
p
I
II
III
IV
1
4
3
2
F
System 12


attempting to start but failing
ppp
I
II
III
IV
1
4
3
2
F
System 13


effortlessly
mp
mf
I
II
III
IV
4
3
1
2
Gb
System 14


moderately fast
pp
I
II
III
IV
3
3
2
C
System 15

HOMENAJE A CERVANTES: FOR VIOLIN, COMPUTER AND PROJECTIONS

Richard Hoadley
rhoadley.net
music@rhoadley.net

ABSTRACT

This is a presentation of the dynamic score *Homenaje a Cervantes* (Homage to Cervantes) created for violin, computer and projections, originally commissioned for and first performed at the University of A Coruna, Spain in May 2017. The piece has been composed using the software packages SuperCollider and INSCORE; the violin part should be played live from a laptop screen or a projection. The texts used are the original Cervantes text, an English translation and a series of original poems created specially for this project by the poet Phil Terry.

1. INTRODUCTION

This paper seeks to demonstrate selected aspects of the technical and aesthetic structure of the composition *Homenaje a Cervantes*; in particular it exposes and examines the layers of code constructed to generate the notations for performance by the violinist. Algorithmic material manipulating audio, text and graphics is generated through scheduling of functions, audio analysis and in certain configurations physical computing or a combination of these elements. Functions and processes are constructed within SuperCollider's native language (**sclang**)[1]. Parts of the piece have been arranged to allow interaction with a dancer should one be available. These parts (primarily the sustained violin material towards the end of the piece) utilise the Microsoft Kinect for Xbox One sensor. SuperCollider algorithms generate time, frequency, amplitude and control values which are then sent either to the SC synth or via Open Sound Control (OSC), (using a custom SC class developed by the author) to the programme INSCORE [2]. INSCORE is able to generate and control a variety of notations, including common practice music notation. While, for both technical and musical reasons, I am currently most involved with the latter aspect, I am involved in other collaborative projects making use of generative graphics and text.

One way in which SuperCollider can be used is by selecting a line or larger region of code and 'evaluating' it. The relevant code is rendered immediately (i.e. just in time, or as soon as possible). If a section of code is enclosed within two parentheses — (*code here*) — inserting the cursor at any point between these and pressing <enter> or equivalent will evaluate that section of code. As the piece is to a significant extent performed through such live evaluation of code segments it can be said to exploit certain live coding practices. Due to the complex construction of much of the code, however, only a minimal quantity of actual typed coding is undertaken during any given performance. Deciding what is coded, when and how during a live coding performance is a fascinating issue worthy of significant future research. There is further discussion of live coding in my own work in [3] and more about the generic idea of live coding in [4].

To fully appreciate what follows some familiarity with SuperCollider's built in language **sclang** and Guido notation is desirable. The code is provided here for illustrative purposes only. It will not run successfully without a variety of dependencies.

At various points in the text I refer to passages from a demonstration video. This can be found at the following web address: http://rhoadley.net/video/homage.

2. TITLE

Figure 2 shows *Homenaje*'s title screen and **code listing 1** shows the relevant 'live' source code. The latter demonstrates an important aspect of my use of INSCORE in this case: the necessity to 'reset' particular groups of elements at strategic moments. In this case, if *Homenaje* has been played or rehearsed and we wish to return to the beginning, many objects will have been displaced and reformatted. The apparently redundant codes here, such as moving, scaling and (re-)setting the origin are necessary for this reason. The code also shows the way in which even in this digital environment, there is still a need for reference to physical aspects of the score (e.g. page height and width).

Lines of code beginning `~homage` reference the INSCORE class for SuperCollider prepared by the author. This is purely a convenience class designed to make the coding of Guido music notation within INSCORE easier and more straightforward. In the following case, the following line of code:

```
~homage.note("homageWin", 0, "a")
```

outputs to INSCORE the following OSC string:

```
/ITL/homageWin/score0 set gmn [a]
```

which INSCORE converts into the notation snippet shown in figure 1. The INSCORE class, once complete, including

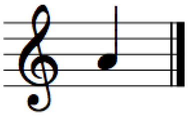


Figure 1: A simple INScore score.

properly prepared SC help files, will be made available for public download [1] . As can be seen in line 19 of **code listing 1**, the class makes use of INSCORE's ability to make use of fully formattable HTML code.

Figure 2: the title screen.

```
1
2  // the code listing below displays the title and formats the
       score
3  (
4  ~pageWidth="28cm";
5  ~pageHeight="36cm";
6  ~myNoteArray = " "; // empty score
7  ~homage.note("homageWin", 0, ~myNoteArray);
8  ~homage.scale("homageWin", "score", 0, 0.5); // size the score
9  ~homage.move("homageWin", "score", 0, -1.4, -0.8, 1); //
       position the score
10 ~homage.origin("homageWin", "score", 0, -1, -1); // position
       the score
11 ~homage.htmlFull("homageWin", "html", 0, "Helvetica", "50pt",
       "normal", "normal", 0.1, ~argbConvert.value([255, 100,
       100, 100]), " ");
12
13 Task({ 0.25.wait;
14    ~homage.move("homageWin", "text", 0, 3.0, 3.0);
15    ~homage.scale("homageWin", "html", 0, 1);
16    ~homage.rotate("homageWin", "html", 0, 0, 0); // due to
       later transformations, make sure the rotation of the text
       is reset
17    ~homage.origin("homageWin", "html", 0, 0, 0);
18    ~homage.move("homageWin", "html", 0, 0, 0);
19    ~homage.htmlFull("homageWin", "html", 0, "Helvetica",
       "50pt", "normal", "normal", 0.1, ~argbConvert.value([255,
       100, 100, 100]), "homenaje a cervantes"); // this is the
       title
20 }).play;
21 )
```

Code listing 1: the title screen listing

3. MELODY

Once the title has faded, sounds of horses walking and the wind blowing emerge, creating an atmospheric sonic backdrop. Blurred coloured areas fade in and out. The colours used (green, blue, grey and brown) are algorithmically generated variants of four prominent colours present in the landscape of La Mancha[5] (see figures 4 and 5, the latter of which shows the background colours in use).

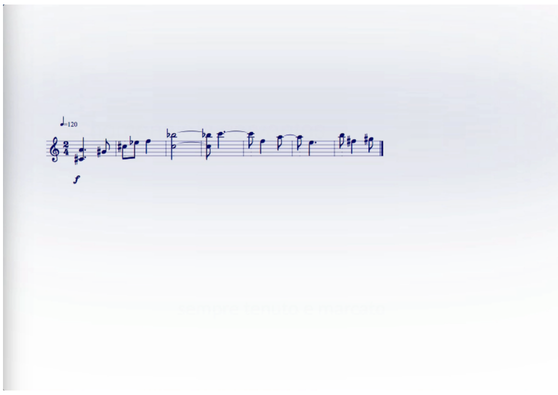


Figure 3: Initial melody.

Figure 4: Windmills and landscape at La Mancha, Spain, showing the colours of the landscape used as backdrops for the notations.

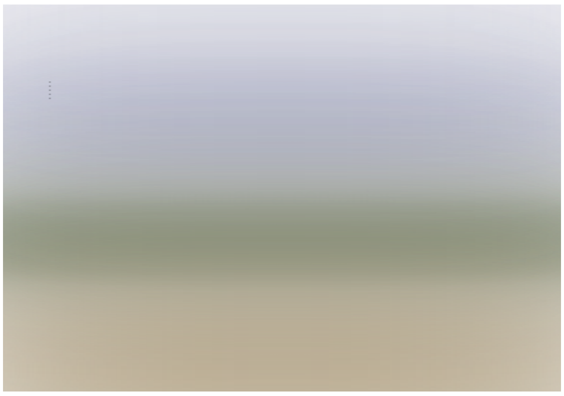


Figure 5: Example of colours from figure 4 used as backdrop

Figure 3 shows a rendition of *Homenaje*'s opening melody, and **code listing 2** the equivalent code. Please refer to the inline comments for more explanatory detail about the code itself.

```

(
```

[1] The current version can be downloaded here: http://rhoadley.net/inscore/INScore.sc , but without documentation

```
3  ~playViolin = true; ~myType = 1; // '~playViolin' set to true
       will play an audio rendition of the melody.  If a real
       violinist is available, this should be set to false.
       '~myType' provides options in the style of the rendition.
       If '~playViolin' is true, then '~myStyle' set to 1 will
       play louder, more vigorous sounding violin samples
4
5  Task({
6      // this is the Task that generates a version of the melody.
       Note that there are still a lot of formatting issues to be
       dealt with as music notation is so predominantly a
       graphic/semantic language.
7
8      ~homage.colour("homageWin", "text", 0, [0,0,0,255]); // set
       the colour
9      4.do({|i| ~homage.text("homageWin", 0, text: 4 - i);
       0.5.wait; }); // generate a count in
10
11     // the three functions using ".stop" end any already running
       function.  We then clear the score and run the melody
       generator (~doViolinMelWithCursorFunc) again.
12     ~colourFadeTask.stop; ~homageViolinMelTask.stop;
       ~homageCursorTask.stop;
13     ~homageViolinMelody = ~scoreFormat ++ " \\intens<\"f\",
       dy=-10hs>";
14
15     // ~doViolinMelWithCursorFunc is the main function, which
       calls on a further function to generate the melody, play
       it, display it and synchronise it with a cursor to aid
       real-time performance
16     ~doViolinMelWithCursorFunc.value(rrand(6, 15), [55, 74], 2,
       [~minMaxAmp[0], ~minMaxAmp[1]], 120, ~myType, ~playViolin,
       durFactor: 8, countIn: 2);
17
18     // the below displays the instruction "sempre tenuto e
       marcato".
19     // both ~doViolinMelWithCursorFunc and ~colourFadeGlobal
       fade the relevant element.
20     ~homage.text("homageWin", 0, text: "sempre tenuto e
       marcato"); ~homage.colour("homageWin", "text", 0,
       [0,0,0,255]);
21     ~colourFadeGlobal.value("~homage", "homageWin", "text", 0,
       [255, 0], 4);
22 }).play; // end of melody generating task.
23 )
```

Code listing 2: Top level code of opening melody

Code listing 2 is top level code. In performance, evaluation of this complete section causes the main work to be done by the function `~doViolinMelWithCursorFunc` (see **code listing 3**). This function synchronises `~doHomageViolinMel` (line 4) with a moving cursor intended to help the performer.

```
1
2  (
3  ~doViolinMelWithCursorFunc = ({ arg noteNum=4, range=[55, 62],
       octaves=2, amplitude=[0.6, 0.8], time=120, type = 1,
       play=true, durFactor=1.0, countIn = 2; // bpm
4      var cursorPosX=0.1, cursorPosY = -0.5;
5
6      // colour the score black
7      ~homageGuido[0][10] = [0,0,0,255];
       ~homage.colour("homageWin", "score", 0, [0,0,0,255]);
8
9      ~homageViolinMelTask.stop; ~homageCursorTask.stop; // stop
       any existing tasks
10     ~homage.scale("homageWin", "score", 0, 0.45); // set the
       scale
11     ~homage.tempo("homageWin", "cursor", 0, time); // set the
       cursor tempo
12
13     // the function that generates the melody itself
14     ~doHomageViolinMel.value(noteNum, range, octaves,
       amp:{amplitude.choose}, type: type, wait: true, play:
       play, sayDone: true, waitFactor: (60/time)*0.5, mm: time,
       scoreSize: ~scoreSize, durFactor: durFactor, countIn:
       countIn);
15
16     ~homage.date("homageWin", "cursor", 0, 0); // the cursor
17
18     // describe the movement of the cursor
19     ~homageCursorTask = Task({
20         var date, noteWait, totalWait=0;
21         noteNum.do({|i|
22             ~homage.position("homageWin", "cursor", 0, cursorPosX,
       cursorPosY);
23             if ( cursorPosY == -0.5, { cursorPosY = -0.7 }, {
       cursorPosY = -0.5 } );
24             noteWait = (60/time);
25             totalWait = totalWait + noteWait;
26             noteWait.wait;
27         });
28
29         (totalWait*0.75).wait;
30
31         ~homageGuido[0][10][3] = 255; // set alpha channel to
       opaque
32         ~colourFadeGlobal.value("~homage", "homageWin", "score",
       0, [~homageGuido[0][10][3], 0], 3); // fade the score
33     }).play;
34 });
35 );
```

Code listing 3: ~doViolinMelWithCursorFunc

The violin melody itself is generated (and optionally either played, displayed or both) by `~doHomageViolinMel`, shown in **code listing 4**. Available arguments include:

- the number of notes;
- the range of the notes in terms of midi pitch;
- the range of the notes in terms of the number of octaves potentially covered;
- the variety of durations to be used and the weightings of those durations;
- the duration and amplitude of each audio rendered note;
- whether to display the score at once or in real time, whether to play it, whether to display it at all.

The process of 'composing' these algorithms is itself an essential part of the creative act: each compositional gesture will require different musical options. Arguments are added (or, more rarely, taken away) as the aesthetic need arises. If a function is used in a subsequent composition it is likely that these arguments will be tidied up in order to promote clarity and ease of use.

```
1  (
2  ~doHomageViolinMel = ({ arg num=10, range=[67, 74], numOct=2,
       durRange=[0,1,2,3,4,5],
       durWeight=[0.15,0.24,0.21,0.21,0.12,0.08], durFactor=1.0,
       amp=1.5, wait=true, play=true, display=true,
       waitFactor=0.25, type=1, halo=[4.0, 10.0], sceneNum=0,
       sayDone=false, mm=120, scoreSize =
       "\\pageFormat<w=20cm,h=24cm>", countIn = 2;
3      var durDictionary= Dictionary.newFrom(List[0, " ", 1,
       "*1/8", 2, "*1/4", 3, "*1/4.", 4, "*1/2", 5, "*5/8", 6,
       "*1/2.", 7, "*7/8", 8, "*1/1", 9, "*9/8"]), score =
       scoreSize + "\\clef<\"treble\"> \\meter<\"2/4\">
       \\tempo<\"[1/4]=" + mm + "\" ,dx=-5, dy=4> _ _", dur = 1,
       note = rrand(range[0], range[1]), prevNote = 0, octave = [
       0, 1 ].choose, chordInt = 0, chordDurString, chordDurNum;
4
5      ~homageViolinMelTask.stop; // stop any existing running tasks
6
7      if ( ~homageViolinMelody != "", { score =
       ~homageViolinMelody } );
8
9      ~homageViolinMelTask = Task({
10         num.do({
11             if ( durWeight == "choose", { dur = durRange.choose},
       { dur = durRange.wchoose(durWeight) } ); // dur is the
       number (1-8)
12
13             octave = numOct.rand; // choose octave
14
15             // if there is a repeated note, get a new one
16             while ( { note == prevNote },
17                 {
18                     note = rrand(range[0], range[1]); // pick a
       pitch within the range
19                     note = note + (octave*12); // transpose the
       pitch to the octave chosen earlier
20                 } );
21
```

```
22          prevNote = note; // keep a record of the chosen note
23
24          // if there's a chord, you have to notate it
       differently (a chord is indicated by setting 'dur' to 0)
25
26          if ( dur == 0, {
27             chordInt = [8, 9, 10].choose; // choose the chord
       interval.  The interval will be a minor 6th, a major 6th
       or a minor 7th as these are relatively straightforward
       intervals for a violinist to play.  If a chord is needed,
       'dur' is set to zero, so we have to pick another duration.
        We pick from the original list, but without the zero
       which we don't want to pick again.
28
29             if ( durWeight == "choose", { chordDurNum =
       durRange.choose}, { chordDurNum =
       durRange.wchoose(durWeight); } );
30
31  // if zero has been chosen, pick something else from the list
       until it isn't zero (the chord still needs a duration)
32             while ( {chordDurNum == 0}, {
33                if ( durWeight == "choose", { chordDurNum =
       durRange.choose}, { chordDurNum =
       durRange.wchoose(durWeight); } );
34             } );
35
36             chordDurString = durDictionary[chordDurNum]; //
       string of the chord duration (i.e. "*1/4")
37
38  // add to the score string, if we are using a chord
39             score = score + "{" ++ ~guidoNoteMap.value(note) ++
       chordDurString ++ "," ++
       ~guidoNoteMap.value(note+chordInt) ++ "}";
40
41          }, { // add to the score string if there *isn't* a
       chord
42             score = score + ~guidoNoteMap.value(note) ++
       durDictionary[dur];
43          } );
44
45          // send to INScore if display is true
46          if ( display == true, { ~homage.note("homageWin",
       sceneNum, score ) } );
47
48          ~homageViolinMelody = score; // store the score data
       in an environment variable
49
50          if ( play == true, { // do we want to hear it?
51             if ( dur == 0, { // if it's a chord
52
53                // for aggressive playback, type = 1
54                if (type == 1 , {
55  ~playViolinArcoSforzando.value(((chordDurNum)*waitFactor)*durFactor,
       0.01, 0.5, 0.49, (note), 1.0, amp);
56  ~playViolinArcoSforzando.value(((chordDurNum)*waitFactor)*durFactor,
       0.01, 0.5, 0.49, (note + chordInt), 1.0, amp); // second
       note
57                },{
58  ~playViolinArcoGentle.value(((chordDurNum)*waitFactor)*durFactor,
       0.01, (note + chordInt), 1.0, amp)
59             } );
60
61  // the 'halo' effect generates a longer (usually quieter)
       sustained note emanating from each `formally' played note,
       creating a `halo' of sound.  To switch off use halo = [0,0]
62                ~playViolinArcoGentle.value(rrand(halo[0],
       halo[1]), 0.01, (note + chordInt), 1.0, amp);
63             });
64
65             if (type == 1 , { // aggressive or gentle?
66
       ~playViolinArcoSforzando.value((dur*waitFactor)*durFactor,
       0.01, 0.5, 0.49, note, 1.0, amp)
67                }, {
68        ~playViolinArcoGentle.value((dur*waitFactor)*durFactor,
       0.01, note, 1.0, amp);
69             } );
70
71        ~playViolinArcoGentle.value(rrand(halo[0], halo[1]),
       0.01, note, 1.0, amp);
72          }); // if play is false, we skip the above
73
74
75          if ( wait == true, { // do we want the whole score at
       once (false, we don't want to wait)?
76             if ( dur == 0, { ((chordDurNum)*waitFactor).wait;
       }, {(dur*waitFactor).wait; } );
77          } );
78       }); // end do
79
80       // tell us if you've finished playing/printing the melody
81       if ( sayDone == true, {
82          "homageViolinMelody done".postln;
83          ~homage.colour("homageWin", "score", 0, [0,0,100,255]);
84          ~homageGuido[0][10] = [0,0,100,255];
85       });
86    }).play;
87  });
```

Code listing 4: ~doHomageViolinMel

3.1 The Aesthetics of Melody Generation

Straightfoward algorithms govern the details of the generation of this initial melody. It terms of compositional process, the ideas are developed in ways that mirror (my own) 'standard', non-digital compositional methods, focusing on traditional musical elements such as atmosphere, tempo, dynamics, articulation and tessitura. The development of the construction of the algorithm relies very much on trial and error, although previous experience gained through the previous development of algorithms has a substantial influence.

In **code listing 4**, one of the first 'decisions' involves the choice of duration for a new event. This involves a variety of durations (line 2):

```
durRange=[0,1,2,3,4,5]
```

and corresponding probability weightings:

```
durWeight=[0.15,0.24,0.21,0.20,0.12,0.08]
```

Because the creative intention was to create short, intensive, active and rhythmically metrical phrases, `durRange` only consists of whole notes, where '1' represents one quaver's duration. As the intended mood of these melodies is vigorously active, the weightings favour the shorter durations with the most weighted duration being one crotchet (one quarter note) in length. Ultimately, the final values for these are achieved through repeated generation and regeneration until the aesthetically required balance is achieved.

Subsequent algorithms choose the octave of the pitch, and the pitch itself, in this case a randomised value from the provided range. A previous duration value of zero indicates that a chord should be created, and as the violinist must navigate the dynamic part at sight, only the most straightforward violin diads are allowed: a minor or major sixth or a minor seventh (**code listing 4**, line 26+).

The algorithm also includes the possibility of specifying one of two different 'styles' of playback (should playback be required in the absence of a human violinist). These styles are either using louder samples with a sharp attack or quieter samples with a more gentle attack.

4. TEXTS, CHORDS AND LINES

The next section of *Homenaje* (from about 2:50 of the demonstration video) introduces sections of text from Cervante's original version of *Don Quixote* [2] as well as an English translation [3]. Text excerpts are accompanied by a variety of musical figures, described as either *lines* or *chords*. Each of these types of figures has originally appeared in earlier compositions (e.g. *Calder's Violin*[6] and *How To Play the Piano*[7]).

4.1 Texts

The digital texts were converted into UTF-8 format text files allowing for the straightforward inclusion of Spanish accents. During the set-up of the piece these files are loaded into environment variables:

[2] Original available here
[3] English translation available here

```
g = File(~path ++
    "text/quijote_sp_text_only_utf-8.txt","r");
~homSpWords = g.readAllString;
g.close;
```

The entire text is then converted to lowercase so avoiding difficulties of sentence construction using upper-case lettering:

```
~stanzaTextInputSp = ~homSpWords.toLower;
```

Only small pieces of the text are chosen for display. Due to the size of the entire text and the time it can take to choose a range within it, a smaller section is chosen in advance, and the smaller chunk to be used is taken from that and stored in another environment variable. In the below, a starting point is chosen from within the entire text, and then a chunk of 1000 characters is chosen from that starting point:

```
// used in ~generateFullStanza
~randPlaceInStream =
    ((~stanzaTextInputSp.size)-1200).rand;
~stanzaTextInputSpSmall =
    ~stanzaTextInputSp[~randPlaceInStream..
(~randPlaceInStream+1000)];
```

Below is the function `~homageTextFunc` which automates this process. The arguments mean that the main function will be run just once, and will produce a 'stanza' of one word and one line. After it has appeared, it will fade out in between 0.2 and 6 seconds. The language chosen will be English (a zero value is Spanish and a value in between is the probability that one language or the other will be chosen — a value of 0.5 will mean each language is equally likely. At the same time a fragment is chosen from audio files of readings of each version. This will have an amplitude of between 0.1 and 0.35 (maximum amplitude is nominally 1.0), and the function will pause between 1 and 4 seconds between lines and 2 and 7 seconds between 'stanzas':

```
~homageTextFunc.value(1, [1, 1], [0.2,
    6.0], 1.0, [0.1, 0.35], [1.0, 4.0],
    [2.0, 7.0]);
```

Code listing 5 contains a full listing of the function.

```
(
~homageTextFunc = ({ arg doNum=40, stanzaDoNum=[1, 6],
      stanzaFadeTime=[0.2, 6.0], langVar=0.5, ampArray=[0.1,
      0.35], lineWait=[1.0, 4.0], stanzaWait=[2.0, 7.0];
   ~stanzaRepeatTask.stop; ~stanzaTask.stop; // stop any
      running tasks
   ~stanzaRepeatTask = Task({
      var lang = ~voiceBufEn;
      doNum.do({ |i|
         ("~homageTextFunc stanza:" + i ++ "/" ++
      doNum).postln; // keep us informed
         (
            ~stanzaWordNum = [3, 6]; ~stanzaLineNum = [1, 3];
            ~stanzaTask = Task({
               var stanzaTaskNum = rrand(stanzaDoNum[0],
      stanzaDoNum[1]);
               stanzaTaskNum.do({ |i|
                  ("stanzaTaskNum:" + i ++ "/" ++
      stanzaTaskNum).postln;

      ~generateFullStanzaOrig.value(rrand(stanzaFadeTime[0],
      stanzaFadeTime[1]));
                  if ( langVar.coin, { lang = ~voiceBufEn }, {
      lang = ~voiceBufSp });
                  // here is the audio taken from readings of
      the book in Spanish and English.  Currently the audio is
      chosen at random from within the files
                  if ( 0.99.coin, { ~fragmentSynthEnv =
      Synth.new("fragmentSynthEnv", [\dur, rrand(3.1, 8.0),
      \amp, rrand(ampArray[0], ampArray[1]), \bufnum,
      lang.bufnum, \startPos, rrand(0, lang.numFrames), \rate,
      1, \attack, 0.3, \sustain, 0.4, \release, 0.3, \effectBus,
      ~effect]); } );
                  rrand(lineWait[0], lineWait[1]).wait;
               });
            }).play;
         );
         rrand(stanzaWait[0], stanzaWait[1]).wait;
      });
   }).play;
});
);
```

Code listing 5: ~homageTextFunc

```
(
// initialise variables
if ( ~stanzaLayerNum == nil, { ~stanzaLayerNum = 0 });
if ( ~stanzaSceneArray == nil, { ~stanzaSceneArray = [
      nil,nil,nil,nil,nil,nil,nil,nil,nil,nil,nil ] }); // this
      array is to keep track of which object layers contain what
      information and whether they are fading, rotating, growing
      or shrinking, etc.
~fullStanzaFontSize=10;

// the function
~generateFullStanzaOrig=({ arg fadeTime=((4.0.rand)+0.5);
   var colourRand=155.rand, alphaRand=(155.rand)+100,
      hexColour, textStreamPortion="", randPlaceInStream,
      fadeInAndOutRoutine;

   // choose a layer that is not already involved (e.g. fading)
   while ( { (~homageHTML[~stanzaLayerNum][12][1] == true) ||
      ~stanzaSceneArray.includes(~stanzaLayerNum) == true }, {
      ~stanzaLayerNum = 12.rand } );

   // set the font size, face, colour and alpha values
   ~homageHTML[~stanzaLayerNum][15] =
      ((~fullStanzaFontSize.rand)+5).asString ++ "pt";
   ~homageHTML[~stanzaLayerNum][14] = ~fontArray.choose;
      hexColour = ~argbConvert.value([alphaRand, colourRand,
      colourRand, colourRand]);
   ~homageHTML[~stanzaLayerNum][10] = [colourRand, colourRand,
      colourRand, alphaRand];
   ~homageHTML[~stanzaLayerNum][11] = alphaRand;

   // choose the larger section
   randPlaceInStream =
      ((~stanzaTextInputSpSmall.size)-120).rand;

   // choose the smaller text from within this
   textStreamPortion =
      ~stanzaTextInputSpSmall[randPlaceInStream..(randPlaceInStream+120)];

   ~stanzaText = ~createTextStreams.value(textStreamPortion,
      ~stanzaWordNum, ~stanzaLineNum);

   ~phonemeSigns[~stanzaLayerNum] = ~stanzaText;

   // this actually sets the text on screen
   ~homage.htmlFull("homageWin", "html", ~stanzaLayerNum,
      ~homageHTML[~stanzaLayerNum][14],
      ~homageHTML[~stanzaLayerNum][15],
      ~homageHTML[~stanzaLayerNum][2],
      ~homageHTML[~stanzaLayerNum][3], 0.1, hexColour,
      ~phonemeSigns[~stanzaLayerNum]);

   // move the text
   ~homageHTML[~stanzaLayerNum][4] = (rrand(~winSize[0].neg,
      ~winSize[0]))*0.6;
   ~homageHTML[~stanzaLayerNum][5] = (rrand(~winSize[1].neg,
      ~winSize[1]))*0.6;
   // set to (slightly smaller than) the window bounds
   ~homage.move("homageWin", "html", ~stanzaLayerNum,
      ~homageHTML[~stanzaLayerNum][4],
      ~homageHTML[~stanzaLayerNum][5], ~stanzaLayerNum);

   // scale the text
   ~homageHTML[~stanzaLayerNum][9] = rrand(4.0, 8.0);
   ~homage.scale("homageWin", "html", ~stanzaLayerNum,
      ~homageHTML[~stanzaLayerNum][9]);

   // now start to fade the text
   ~htmlFadeFull.value("~homage","homageWin", "html",
      ~stanzaLayerNum, ~homageHTML[~stanzaLayerNum][14],
      ~homageHTML[~stanzaLayerNum][15],
      ~homageHTML[~stanzaLayerNum][2],
      ~homageHTML[~stanzaLayerNum][3], 0.1,
      ~phonemeSigns[~stanzaLayerNum],
      [~homageHTML[~stanzaLayerNum][11], 0],
```

```
        ~homageHTML[~stanzaLayerNum][10], fadeTime);

    // collect more than one to avoid repetition when the
        actions overlap...
    ~stanzaSceneArray = ~stanzaSceneArray.add(~stanzaLayerNum);
    if ( ~stanzaSceneArray.size > 10, { ~stanzaSceneArray =
        ~stanzaSceneArray.drop(1) } );
});
)
```

Code listing 6: ~generateFullStanzaOrig

4.2 Lines and Chords

Lines and chords are two types of generative musical figures both of which have appeared in earlier compositions. More detail about lines and chords, their construction, and the manner of their representation can be found in [7].

5. WINDMILLS

One of the main focuses of *Homenaje* must be, of course, La Mancha's infamous windmills (see figure 4). I had discussed with Phil Terry whether he might have any ideas regarding the text of *Don Quixote* and he composed a series of twelve 'quixotes', each of which took a famous scene from the book and created a 'concrete' poem from each in the shape of a windmill. Below is the text of the first one:

Wicked breed
Unimaginable adventures
Monstrous giants
Happy memory

"What giants?"
Said Sancho Panza.
"Those giants that you
can see over there"

replied his master
"with long arms".

Great service, raw novice, arduous combat, cowardly creatures[4]

We made audio recordings of Phil reading out each of the poems and these recordings accompany the concrete visualisations during performance.

Figure 9 shows its implementation visually in INSCORE. This demonstrates more of the text-based features of the programme. In the demonstration video the passage begins at about 4:40.

[4] The scene to which this refers can be found in Book 1, Chapter 8, available here

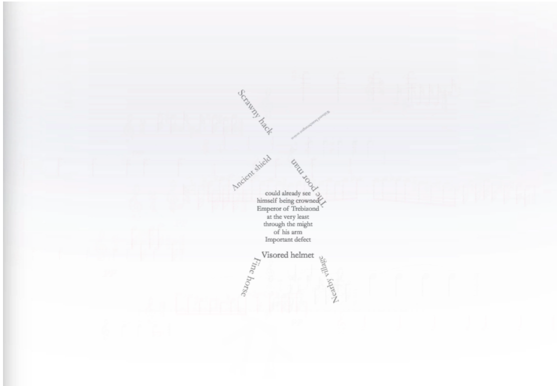


Figure 9: Quixotes: concrete poetry by Phil Terry.

Code listing 7 is rather lengthy, combining as it does complex graphical placements of textual elements.

```
~quixotes = ({ arg windmillNum = 1, num,
      wmSpeedOffsetMinMax=[0.35, 0.45], dilapidation=10,
      scale=1.0, xOffset=0.5, yOffset=0.5;

      // font size needs to be mapped onto the number of
      characters in each word or line so that the cosmetic
      appearance of the windmill graphics can be maintained.
    var font = "Garamond", text, fontSize, fontSizePt,
      fontSizeSpec = ControlSpec(26, 10, 'lin', 1, 20),
      wmSpeedOffset = [ rrand(wmSpeedOffsetMinMax[0],
      wmSpeedOffsetMinMax[1]), rrand(wmSpeedOffsetMinMax[0],
      wmSpeedOffsetMinMax[1]), rrand(wmSpeedOffsetMinMax[0],
      wmSpeedOffsetMinMax[1]), rrand(wmSpeedOffsetMinMax[0],
      wmSpeedOffsetMinMax[1]) ], windmillVarNum =
      ((windmillNum-1) * 14), var windmillTask;

    ~homageHTML.size.do({ |i|
      ~homageHTML[i][9] = scale; // define scale (html 9)
      ~homage.scale("homageWin", "html", i, ~homageHTML[i][9]);
    });

    // windmill blades (lines 0 - 3) text and size
    4.do({|i|
      ~homageHTML[i+windmillVarNum][13] =
      ~quixotesArray[num][i];
      fontSize =
fontSizeSpec.map((~homageHTML[i+windmillVarNum][13].size/12.0)-1.0);
      ~homageHTML[i+windmillVarNum][17] = fontSize.asString ++
      "pt";
      fontSizePt = ~homageHTML[i+windmillVarNum][17];
      ~homage.htmlFull("homageWin", "html", i+windmillVarNum,
      font, fontSizePt, text: ~homageHTML[i+windmillVarNum][13]);
      // move them
      ~homage.move("homageWin", "html", i+windmillVarNum,
      ~homageHTML[i+windmillVarNum][4] + yOffset * scale;,
      ~homageHTML[i+windmillVarNum][5] + xOffset * scale);
    });

    // middle lines text size
    7.do({ |i|
      ~homageHTML[i+4+windmillVarNum][13] =
      ~quixotesArray[num][i+4];
      ~homageHTML[i+4+windmillVarNum][17] = "18pt";
      ~homage.htmlFull("homageWin", "html", i+4+windmillVarNum,
      font, ~homageHTML[i+4+windmillVarNum][17], text:
      ~homageHTML[i+4+windmillVarNum][13]);
    });

    // last three lines text size
    3.do({|i|
      ~homageHTML[i+11+windmillVarNum][13] =
      ~quixotesArray[num][i+11];
      fontSize =
fontSizeSpec.map((~homageHTML[i+11+windmillVarNum][13].size/12.0)-1.0);
      ~homageHTML[i+11+windmillVarNum][17] = fontSize.asString
      ++ "pt";
      fontSizePt = ~homageHTML[i+11+windmillVarNum][17];

      ~homage.htmlFull("homageWin", "html",
      i+11+windmillVarNum, font, fontSizePt, text:
      ~homageHTML[i+11+windmillVarNum][13]);
    });

    // position of body
    if ( dilapidation == 0, { 7.do({|i|
      ~homageHTML[i+4+windmillVarNum][7][2] = 0; }) });

    // note the role of dilapidation here.  Each time the
      function is run during performance, the value of
```

```
    dilapidation increases and so the windmills gradually
    become more and more uneven.
    7.do({ |i|
      ~homage.move("homageWin", "html", i+4+windmillVarNum,
      ~homageHTML[i+4+windmillVarNum][4] + yOffset +
      rrand(dilapidation.neg*0.005, dilapidation*0.005) * scale,
      ~homageHTML[i+4+windmillVarNum][5] + xOffset +
      rrand(dilapidation.neg*0.005, dilapidation*0.005) * scale);
    });

    // create and rotate the last three lines...
    ~homage.rotate("homageWin", "html", 11+windmillVarNum, zPos:
      0);
    ~homage.rotate("homageWin", "html", 12+windmillVarNum, zPos:
      110);
    ~homage.rotate("homageWin", "html", 13+windmillVarNum, zPos:
      250);

    // and move them
    3.do({|i|
    ~homageHTML[i+11+windmillVarNum][4] =
      ~homageHTML[i+11+windmillVarNum][4];
    ~homage.move("homageWin", "html", i+11+windmillVarNum,
      ~homageHTML[i+11+windmillVarNum][4] + yOffset * scale,
      ~homageHTML[i+11+windmillVarNum][5] + xOffset * scale);
    });

    // spin the blades
    windmillTask = Task({
      var zPos1 = (rrand(1.75, 2.25)*wmSpeedOffset[0]);

      while ( { ~homageHTML[0+windmillVarNum][12][3] == true },
      {
          ~homage.drotate("homageWin", "html", 0+windmillVarNum,
      zPos: zPos1);
          ~homage.drotate("homageWin", "html", 1+windmillVarNum,
      zPos: (rrand(1.75, 2.25)*wmSpeedOffset[1]));
          ~homage.drotate("homageWin", "html", 2+windmillVarNum,
      zPos: (rrand(1.75, 2.25)*wmSpeedOffset[2]));
          ~homage.drotate("homageWin", "html", 3+windmillVarNum,
      zPos: (rrand(1.75, 2.25)*wmSpeedOffset[3]));
          0.04.wait;
      } );
    }).play;
});
```

Code listing 7: ~quixotes

Figure 10: ~quixotes showing increasing dilapidation.

Figure 11: ~quixotes flying off.

5.1 Dilapidation

One of the more expressive arguments of the function `~quixotes` (**code listing 7**) is `dilapidation`. This value determines the 'tidiness' of the windmills generated and, reflecting Don Quixote's deteriorating hold on reality, each time the function is called the value is increased and as a consequence the windmills' movements become increasingly uneven and unpredictable. Figure 10 shows two windmills demonstrating this. After a certain level of dilapidation is reached in performance, the windmills disintegrate completely (see Figure 11) and 'fly off' the 'page'.

As dilapidation increases, so too does the disintegration Phil Terry's reading of his Cervantes-inspired poems. With each verse the audio is rendered using decreasing bit and sample rates, making it increasingly incomprehensible. The effects of dilapidation are clearly visible and audible from about 6:00 in the demonstration video.

6. PICASSO

The final section of *Homenaje* features Picasso's 1955 black on white sketch of Don Quixote, his horse Rocinanate, his sidekick Sancho Panza as well as a number of windmills. The drawing was made on August 10, 1955 for the August 18-24 issue (No. 581) of *Les LETTRES françaises*, a weekly French journal directed by Aragon, in celebration of the 350th anniversary of the publication of Don Quixote, Part I [8].

The image has been cut into 23 pieces, some of which can be seen in figure 12. These pieces appear and fade along with the musical algorithms used. In the case of this scene, the musical ideas are initially based around guitar samples (along with occasional harp samples), reflecting the importance of the guitar in Picasso's output: in 2011 there was an exhibition at MOMA — *Picasso: Guitars 1912-1914* (here: https://www.moma.org/calendar/exhibitions/1088). The music consists of a variety of idiomatic guitar ideas and gestures: mordents and turns, strumming, and plucked melody lines (see **code listing 8**), which combine and build into a rhythmic texture (see about 7:40 in the demonstration video and **code listing 9**).

```
// a single guitar note
```

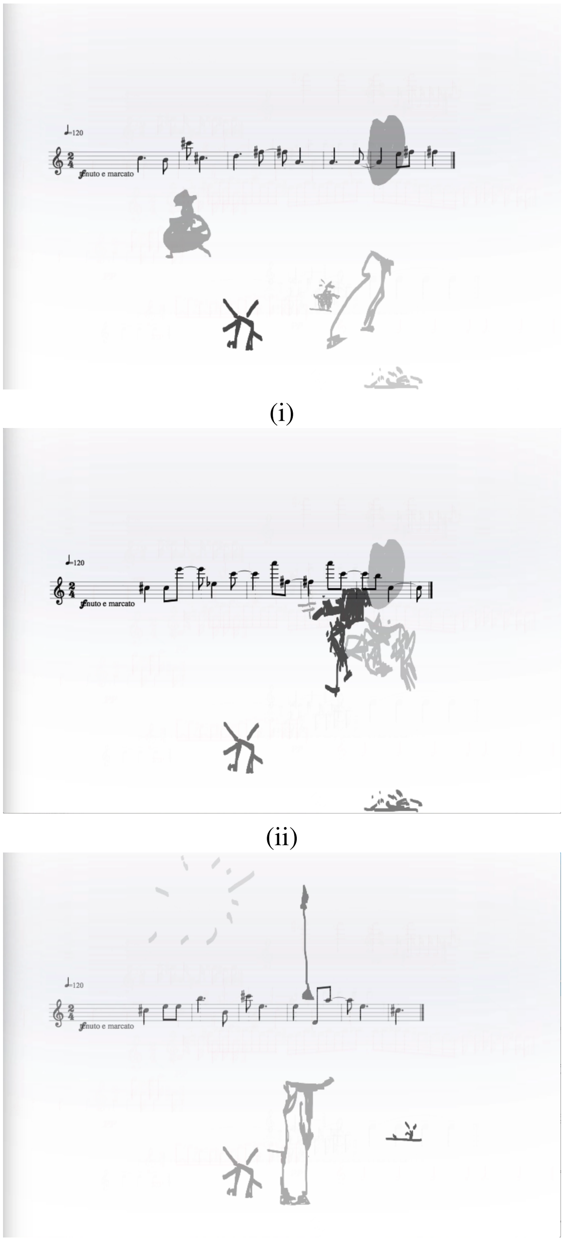
```
// throughout this section the amplitude of the guitar is
      governed by the environment variable ~guitarAmp which
      makes it easier to balance the sound in performance
      environments with different acoustics.

1.do({ ~guitarNote = ([ 50, 57 ].choose)-12;
      ~playGuitar.value(2.1, 0.01, ~guitarNote, 1.0,
      ~guitarAmp); }); ~picassoPartAppearFunc.value(rrand(0.001,
      0.01), 1, false, false);

1.do({ ~guitarNote = ([ 50, 57, 62, 64, 67, 69, 71, 73
      ].choose)-12; ~playGuitar.value(2.1, 0.01, ~guitarNote,
      1.0, ~guitarAmp); });
      ~picassoPartAppearFunc.value(rrand(0.001, 0.01), 1, false,
      false);

// a few notes
// included are functions ~picassoPartAppearFunc and
      ~picassoPartAppearNoFadeFunc which manage the appearance
      and fading of the divided Picasso sketch
// the guitar notes are taken from a central array of values
(
Task({
   var numNotes = rrand(4, 8);
   var waitTime = rrand(0.05, 0.13);

   numNotes.do({ |i|
      1.do({ ~guitarNote = ([ 50, 57, 62, 64, 67, 69, 71, 73
      ].choose)-12; ~playGuitar.value(2.1, 0.01, ~guitarNote,
      1.0, ~guitarAmp); });
      if ( i == (numNotes-1), {
         ~picassoPartAppearFunc.value(rrand(0.001, 0.01), 1,
      false, false);
      }, {
         ~picassoPartAppearNoFadeFunc.value(2.0, 1, false,
      false, rrand(100, 255), 0);
      } );
      (waitTime*(rrand(0.95, 1.05))).wait;
   });
}).play;
);


// low strum
4.do({ ~guitarNote = ([ 50, 57, 62, 64, 67, 69, 71, 73
      ].choose)-24; ~playGuitar.value(2.1, 0.01, ~guitarNote,
      1.0, ~guitarAmp); });
      ~picassoPartAppearFunc.value(rrand(0.001, 0.01), 1, false,
      false);

// arg octave = 0, amp = 3.0;
~guitarStrumFunc.value(2, ~guitarAmp);

// a variety of twists and turns
~guitarTwistFunc.value(2, ~guitarAmp*0.25, true);
      ~picassoPartAppearFunc.value(rrand(0.005, 0.05), 1, false,
      false);
~guitarTwistFunc.value(6, ~guitarAmp*0.25, false);
      ~picassoPartAppearFunc.value(rrand(0.005, 0.05), 1, false,
      false);
~guitarTwistFunc.value(12, ~guitarAmp*0.25, false);
      ~picassoPartAppearFunc.value(rrand(0.005, 0.05), 1, false,
      false);

// many turns
Task({ rrand(2, 8).do({ ~guitarTwistFunc.value(12,
      ~guitarAmp*0.25, false); rrand(0.1, 0.4).wait;
      ~picassoPartAppearFunc.value(rrand(0.001, 0.01), 1, false,
      false); }); }).play;

Task({ 4.do({ ~picassoPartAppearFunc.value(rrand(0.001, 0.01),
      1, false, false); ~guitarTwistFunc.value(12,
      ~guitarAmp*0.3, false); rrand(0.1, 0.4).wait; }); }).play;

Task({ rrand(4, 12).do({ ~guitarTwistFunc.value(12,
      ~guitarAmp*0.25, false); rrand(0.1, 0.4).wait; }); }).play;

~playGuitar.value(0.1, 0.01, 55, 1.0, ~guitarAmp);
      ~picassoPartAppearFunc.value(rrand(0.001, 0.01), 1, false,
      false);

~playTinyGuitarStream.value(rrand(4, 24), true,
      {rrand(~guitarAmp*0.01,~guitarAmp)} );
      ~picassoPartAppearFunc.value(rrand(0.001, 0.01), 2, false,
      false);

~playTinyGuitarStream.value(rrand(4, 24), true,
      {rrand(~guitarAmp*0.01,~guitarAmp)});

// guitar 'strum' function in funcs
~guitarStrumFunc.value(14, ~guitarAmp);

// a small, quiet, relatively metrical guitar melody:
~playTinyGuitarStream.value(rrand(4, 24), true);
      ~guitarStrumFunc.value(2, amp: ~guitarAmp);

// play short burst of guitar melody
(
Task({

   rrand(4, 12).do({
      ~guitarNote = [ 50, 57, 62, 64, 67, 69, 71, 73 ].choose;
      ~playGuitar.value(2.1, 0.01, ~guitarNote, 1.0,
      ~guitarAmp);
      ~picassoPartAppearFunc.value(rrand(0.001, 0.01), 2,
      false, false);
   });

   rrand(4, 18).do({
      ~playGuitar.value(10.5, 0.01, ([ 50, 57, 62, 64, 67, 69,
      71, 73 ].choose)-12, 1.0, ~guitarAmp);
      rrand(0.125, 0.13).wait;
      ~picassoPartAppearFunc.value(rrand(0.001, 0.01), 2,
      false, false);
   });
}).play;
);

// harp
(
Task({

   rrand(4, 12).do({
      ~guitarNote = [ 50, 57, 62, 64, 67, 69, 71, 73 ].choose;
      ~playHarp.value(2.1, 0.01, ~guitarNote, 1.0, 0.4);
         ~picassoPartAppearFunc.value(rrand(0.001, 0.01), 1,
      false, false);
   });

   rrand(4, 18).do({
      ~playHarp.value(10.5, 0.01, ([ 50, 57, 62, 64, 67, 69,
      71, 73 ].choose)-12, 1.0, 0.4);
      rrand(0.125, 0.13).wait;
         ~picassoPartAppearFunc.value(rrand(0.001, 0.01), 1,
      false, false);

   });
}).play;
)

// guitar and harp together
(
Task({

   rrand(8, 18).do({
      ~guitarNote = [ 50, 57, 62, 64, 67, 69, 71, 73 ].choose;
      ~playGuitar.value(2.1, 0.01, ~guitarNote, 1.0,
      ~guitarAmp);
   });

   rrand(8, 18).do({
      if ( 0.25.coin, {
         ~playGuitar.value(10.5, 0.01, ([ 50, 57, 62, 64, 67,
      69, 71, 73 ].choose)-12, 1.0, ~guitarAmp);
      }, {
         ~playHarp.value(10.5, 0.01, ([ 50, 57, 62, 64, 67, 69,
      71, 73 ].choose)-12, 1.0, 0.5);
      } );
      rrand(0.125, 0.13).wait;
   });
}).play;
)
```

Code listing 8: Picasso and guitars 'live coding'

```

// then automated and strictly in time
(
~guitarTranspose = 0;
~guitarOctave = 0;
~picassoCol = [100, 255];

~autoGuitarTask = Task({
   var myWait = 0.125;

   ~autoGuitarTask.stop;
   ~strumTask.stop;

      ~strumTask = Task({ // the initial 'strum'
         rrand(4, 10).do({
            ~guitarNote = [ 50, 57, 62, 64, 67, 69, 71, 73
      ].choose;
            ~playGuitar.value(2.1, 0.01,
      ~guitarNote+~guitarTranspose+~guitarOctave, 1.0,
      ~guitarAmp);
            if ( 0.9.coin, {
               ~picassoPartAppearFunc.value(rrand(0.01, 0.01),
      16, false, false, ~picassoCol);
            }, { ~picassoPartAppearFunc.value(rrand(0.1, 1.2),
      16, false, false, ~picassoCol);
            } );
         });

         rrand(16, 36).do({
            ~playGuitar.value(10.5, 0.01, (([ 50, 57, 62, 64,
      67, 69, 71, 73
      ].choose)-12)+~guitarTranspose+~guitarOctave, 1.0,
```

```
        ~guitarAmp);

                if ( 0.9.coin, {
                    ~picassoPartAppearFunc.value(rrand(0.01, 0.01),
        16, false, false, ~picassoCol);
                }, { ~picassoPartAppearFunc.value(rrand(0.1, 1.2),
        16, false, false, ~picassoCol);
                } );
                myWait.wait;
            });
        }).play;
        myWait.wait;
}).play;
)
```

Code listing 9: The guitar texture algorithm



(i)

(ii)

(iii)

Figure 12: Three renditions of Picasso, guitars and melody.

The guitar-sound-based texture is then augmented by violin melodies, to be played by the live violinist, based on the original melody developed earlier in the piece, but now using the tonality of the rhythmic guitar texture as a tonal basis. These melodies gradually take over the texture, becoming more and more sustained as the guitar fades (from about 10:30, see **code listing 10**). The notes from which those played are chosen are provided by the array

```
~chordNotes
```

which can be coded live, or taken from a set of arrays (see code listing 10) based on the chords shown in figure 13. Harmonically, they comprise fourths and fifths with a more complex overtones and harmonics.

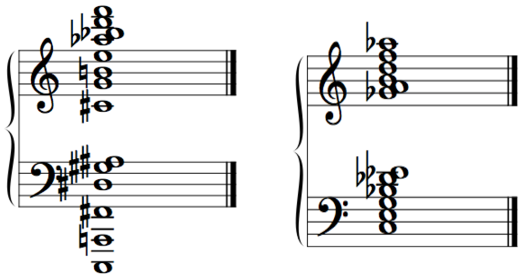


Figure 13: Two tonally ambiguous chords forming the basis of some of the harmonies of the 'sustained string' section of *Homenaje*

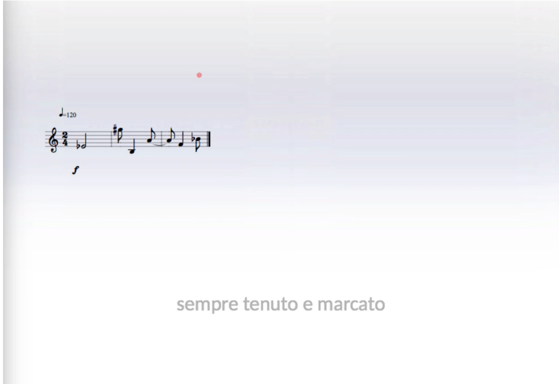
```

// chord Notes
~chordNotes = [ 50, 57, 62, 64, 67, 69 ]; // d, a, d, e, g, a
~chordNotes = [ 50, 57, 62, 64, 67, 69, 71 ]; // + b
~chordNotes = [ 50, 57, 62, 64, 67, 69, 71, 73 ]; // + c#

~chanceOfNote = 0.8;

~chordNotes = [ 50, 57 ]; // fifth
~chordNotes = [ 50, 57, 64 ]; // two fifths
~chordNotes = [ 50, 57, 64, 71 ]; // three fifths
~chordNotes = [ 50, 57, 64, 67, 71 ]; // three fifths

~chordNotes = [ 50, 57, 61, 64, 66, 68, 78 ];
~chordNotes = ~chordNotes + 8
~chordNotes = ~chordNotes - 8
~chordNotes = [ 43, 50, 57, 64, 71 ];
~chordNotes = [ 43, 50, 57, 61, 64, 66, 68, 78 ];

~chanceOfNote = 0.7;

~chordNotes = [ 37, 43, 50, 57, 66, 73, 80, 83, 87 ]; // d, a,
      f#, c#, g#, b, d#
~chordNotes = [ 50, 57, 66, 73, 80, 83 ];
~chordNotes = [ 50, 57, 66, 73, 80 ];

~chanceOfNote = 0.6;

~chordNotes = [ 50, 57, 66, 73 ]; // d, a, f#, c#
~chordNotes = [ 66, 74, 73 ];
~chordNotes = [ 73 ];
```

Code listing 10: Sustained string texture at end

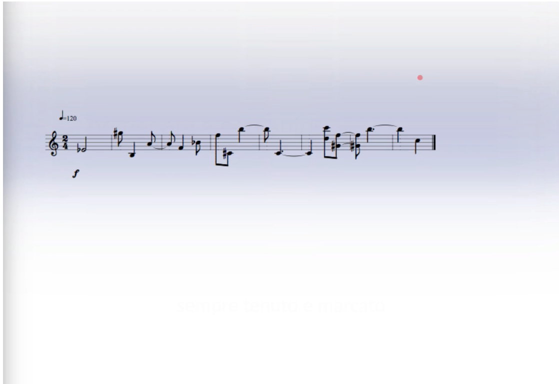
At this point the Picasso sketch is complete and its parts fade in and out, rotating, until they disappears altogether along with the violin music.

7. CONCLUSIONS

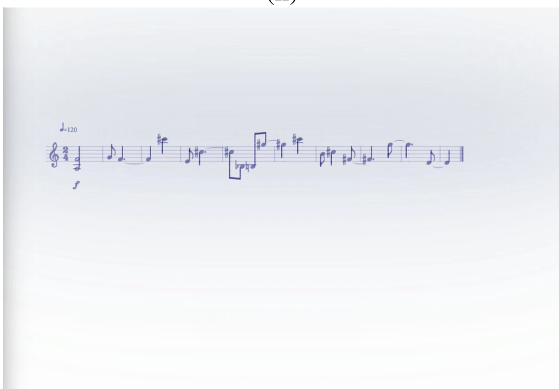
Homenaje is a composition that consciously extends the integration of text and image with music, a process that be-



(i)



(ii)



(iii)

Figure 6: Other renditions of the melody are shown here. Note that in (iii) the melody changes colour from black to dark blue when the fade out begins to indicate this to the instrumentalist.

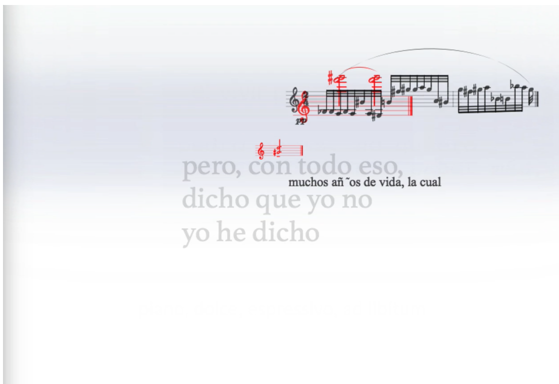



Figure 7: A rendition of text, lines and chords.

gan with the music-dance-text piece *Choreograms* in which 18th century dance notations appeared alongside contemporary poetry, audio and music notations. It formalises the structures put in place in earlier pieces, most notably the sets of object arrays that allow the storage of each objects state (e.g. location, colour, alpha value, rotation).

Each project spent composing with SuperCollider and INSCORE convinces me of the flexibility and power of these pieces of software, both individually and in combination. Each piece of software allows virtually complete control not only over the tools and mechanisms involved, they also allow maximum freedom over how they themselves can be controlled. Both SuperCollider and INSCORE can be used as standalone pieces of software, or as engines to be controlled from other preferred resources. The fact that **sclang** is able to deal so effectively with rather obscure and arbitrary text-based functionality such as converting text to all lower (or all upper, or any number of other string transformations) demonstrates both its flexibility and the importance of that flexibility in cross-domain work. There is little doubt that all of these factors make the investigation of links and mappings between diverse expressive domains particularly suitable for these resources and there is enormous potential for discovery and expression.

8. REFERENCES

[1] S. Wilson, D. Cottle, and N. Collins, Eds., *The SuperCollider Book*. Cambridge, MA: MIT Press, 2011.

[2] D. Fober, Y. Orlarey, and S. Letz, "Inscore – an environment for the design of live music scores," in *Proceedings of the Linux Audio Conference*, 2012, pp. 47–54.

[3] R. Hoadley, "Live coding, live notation, live performance," in *Electronic Visualisation and the Arts (EVA 2016)*, ser. Electronic Workshops in Computing (eWiC), J. P. Bowen, G. Diprose, and N. Lambert, Eds., British Computer Society. British Computer Society, 2016, pp. 34–41.

[4] N. Collins, A. McLean, J. Rohrhuber, and A. Ward, "Live coding in laptop performance," *Organised Sound*, vol. 8, no. 3, pp. 321–330, 2003.

[5] Google, "Google map," June 2017. [Online]. Available: https://goo.gl/maps/EmhqzZdmeA52

[6] R. Hoadley, "Calder's violin (performance)," in *Proceedings of the International Computer Music Conference*. Athens, Greece: ICMA, September 2014.

[7] ——, "How to play the piano," in *Proceedings of the ICMC*, H. Timmermans, Ed., ICMA. Utrecht, Netherlands: HKU University of the Arts Utrecht, 2016, pp. 176–180.

[8] A. G. L. Ré, "A possible source for picasso's drawing of don quixote," *Cervantes: Bulletin of the Cervantes Society of America*, vol. 12, no. 1, pp. 105–110, 1992.

Index of Authors