

SONIFICATION OF SYMBOLIC MUSIC IN THE ELVIS PROJECT

R. Michael Winters & Julie E. Cumming

Schulich School of Music
McGill University & CIRMMT, Montréal, QC
Raymond.Winters@mail.mcgill.ca

ABSTRACT

This paper presents the development of sonification in the ELVIS project, a collaboration in interdisciplinary musicology targeting large databases of symbolic music and tools for their systematic analysis. An sonification interface was created to rapidly explore and analyze collections of musical intervals originating from various composers, genres, and styles. The interface visually displays imported musical data as a sound-file, and maps data events to individual short, discrete pitches or intervals. The user can interact with the data by visually zoom in, making selections, playing through the data at various speeds, and adjusting the transposition and frequency spread of the pitches to maximize acoustic comfort and clarity. A study is presented in which rapid pitch-mapping is applied to compare differences between similar corpora. A group of 11 participants were able to correctly order collections of sonifications for three composers (Monteverdi, Bach, and Beethoven) and three presentation speeds (10^2 , 10^3 , and 10^4 notes/second). Benefits of sonification are discussed including the ability to quickly differentiate composers, find non-obvious patterns in the data, and ‘direct mapping’. The interface is made available as a MacOSX standalone application written in SuperCollider.

1. INTRODUCTION & MOTIVATION

Drawing the boundaries between sonification and music has thus far been a rewarding pursuit, helping identify similarities [1] and clarify differences [2]. However, the two will always have at least one attribute in common: the fundamental medium for display, communication or expression is sound.

The emergence of large databases of music has created an opportunity for sonification to be applied to representing musical data. Examples of such data might be symbolic information such as notes, durations, chords, instruments, but also may be more directly derived audio features, for instance, the spectral centroid, entropy, or flux. Sound is a well-equipped medium for conveying this information which, as some have argued [3], would seem to make sonification a clear choice for researchers working with musical data.

However, overwhelmingly sound is not used as an integrated tool for in large-scale music analysis. All but absent in relevant fields, its appearance is often tacit—displaying the musical outcome of presented algorithms (e.g. [4]). As Table 1 displays, a

word search of the past four conferences in Music Information Retrieval (ISMIR) reveals that words stemming from “sonif-” appear a mere four times. Interestingly, the number of occurrences of words stemming from “listen-” have doubled. For a field where sound is already the essential medium for analysis, and the act of listening equally well-founded, creating sonification applications for music analysis may be an unusually apt prospect.

Table 1: A table of word occurrences in the annual ISMIR conference from 2010-13.

ISMIR	2010	2011	2012	2013
“Sonif-”	0	3	1	0
“Listen-”	234	365	485	460
# of pages	688	826	642	644

This paper presents the development of sonification in the ELVIS project, an interdisciplinary collaboration in digital musicology. A sonification interface for exploring musical is described targeting intervals derived from large databases of music from the 15th to 19th centuries. Imported musical data is displayed visually as a sound-file and mapped to sound using individual, short discrete pitches or intervals. The GUI allows the user to interactively select a broad range of playback speeds, and control elements of the mapping to maximize clarity and comfort. A study is presented in which rapid pitch-mapping is applied towards comparing differences in pitch content of similar corpora. Results from a test involving 11 participants demonstrate that the technique could be used to correctly order sonifications by number of differences for three composers (Monteverdi, Bach, Beethoven), and three presentation speeds (10^2 , 10^3 , and 10^4 notes/second). Benefits of sonification are discussed including the ability to differentiate composers, hear non-obvious data features, and ‘direct mapping’.

2. BACKGROUND

2.1. Introduction to ELVIS

The ELVIS project is an interdisciplinary collaboration in digital musicology combining music historians, theorists, musicologists, technologists, and computer scientists. The group has amassed a large database of symbolic music from the 15th to 19th centuries that has been made searchable and publicly available online (elvis-project.ca) To analyze this music, there are a variety of computational tools available, but none which were developed for the express purpose of contrapuntal analysis—the ‘horizontal’ and ‘vertical’ movement of individual musical voices in a polyphonic musical structure. The ELVIS team created such a tool called ‘VIS’,



made its simple operations accessible through a web-application, and the more advanced features available through working with the freely available python-based API [5]. VIS uses music21 [6], a more established python-based music analysis framework, to assist in processing data.

Using music21 and VIS, it is possible to amass musical information derived from large and sometimes complete collections of music from the 15th to 19th centuries. Such corpora contain substantial amounts of data—for instance the collection of all pitches and durations from the Beethoven String Quartets or the individual intervals between pitches simultaneous or adjacent (in-time) pitches. Though the possibilities for derived data are numerous, pitch and interval content are among the most important in analyzing western art music, and can lead to important descriptions of composer, genre, and style.

2.2. The Choice of Sonification

To explore these large databases, the ELVIS project pursued sonification as an alternative tool for data analysis. Previous work had demonstrated a rapid pitch-mapping technique capable of creating characteristic sounds of individual composers and genres while drawing attention to more local events and structures [7]. The ELVIS team chose to incorporate sonification for these data exploration qualities, but also welcomed sound as an apt medium for displaying musical data (i.e. ‘direct mapping’ Sec. 5.2). During the course of collaboration in the ELVIS project, the analysis capacity of the original technique was strengthened through integration in a graphical user interface including interactive controls of both the sonification mapping and visual display.

3. THE SONIFICATION INTERFACE

The sonification interface was written in SuperCollider [8] using the QtGUI framework. QtGUI includes a SoundFileView, which as applied in this context, presents the user with imported data as if it were a sound-file itself. When the user presses the ‘play’ button, a visual cursor follows the exact position in the data that is being sonified. When a user hears something in the data that had not before been visually obvious, the user can visually zoom-in to the data using a mouse. To record the finding, the user can make a selection of the data (as demonstrated in Fig. 3) and press the ‘record’ button in the upper right-hand corner of the interface. The interface plays through the selected data and saves the resultant sound as a WAVE file, which by default is saved to the desktop.

The interface accepts a two-column CSV data type, which in the case of ELVIS, represents the collection of all vertical and horizontal intervals between all voices in a polyphonic work in order of occurrence for each voice. It assumes the first column is vertical intervals and the second column are the horizontal intervals. It uses this information to display horizontal and vertical intervals as ‘two-tracks’ in the SoundFileView. The user can choose which track is being played by selecting from the ‘vertical’ and ‘horizontal’ checkboxes to the immediate right of the ‘play’ button in Fig. 3

3.1. Mapping Strategy

After data is imported, two mapping strategies are offered to the user and are selectable using a pop-up menu:

1. Intervals mapped to pitches

2. Intervals mapped to intervals

If the first option is selected, each interval in the imported data is played as a single pitch in a chromatic scale (e.g. $1 \rightarrow C\sharp$, $4 \rightarrow E4$, $5 \rightarrow F4$, $-3 \rightarrow A3$) In the second, each interval is represented as a pair of consecutive ascending or descending diatonic intervals beginning with the unison and expanding pseudo-symmetrically with increasing magnitude (e.g. $1 \rightarrow \{C4, C4\}$, $2 \rightarrow \{C4, D4\}$, $3 \rightarrow \{B3, D4\}$, $5 \rightarrow \{A3, E4\}$, $-3 \rightarrow \{D4, B3\}$).

In both cases, the individual pitches are represented by sinusoids whose frequencies are determined by interpreting the data as a MIDI note value, transposing upwards by a user-defined value (See ‘Pitch’ knob in Sec. 3.2), and converting to cycles per second using the .midicps method in SuperCollider. For most playback speeds, the sinusoids are shaped using a 50ms duration sinusoidal envelope as shown in Fig. 2. The choice of amplitude envelope and duration was made to facilitate high-speed playback. For durations shorter than 50ms, pitches became noise-like and pitch could not be perceived. For other envelopes, ‘clipping’ and other artifacts were audible. The resulting overlap between notes was therefore minimized though not negligible, varying with speed: 5 note overlap at 10^2 notes/second, 50 note overlap at 10^3 notes/second, and 500 note overlap at 10^4 notes/second.

The amplitudes of individual pitches were also modified to be roughly equal in loudness using the AmpCompA unit generator with MIDI number 40 (E2) as reference. The algorithm uses psychoacoustic equal-loudness contours [9] to compensate for the fact that pitches in the range of 1.5-7kHz will sound louder than other pitches. For sonification, it was desired that all pitches would be of roughly equal loudness as the reference tone.

3.2. Sonification Controls

When sonifying data, the interface supports three acoustic adjustments to the mapping strategy:

Spread Pitch range can be expanded or contracted.

Pitch The center frequency can be transposed up or down.

Speed The playback speed can be increased or decreased.

The choice of names for the three knobs (‘Spread’, ‘Pitch’, ‘Speed’), though admittedly un-descriptive, were chosen to effectively communicate the nature of each control to an audience that may not be familiar with sound synthesis.

The ‘Spread’ knob can be adjusted from ‘0 to 2 Octaves’ meaning that the span of an octave in the sonification can be contracted to the unison or expanded to two octaves. The default value of ‘1 Octave’ means that there is no alteration. This control may be used to ‘spread out’ the pitch range of a sonification. For instance, if the input data were limited to the range of -8 to 8 (descending/ascending octave)—as the horizontal intervals in renaissance composers tend to be—adjusting the spread to ‘2 Octaves’ would spread the input data from -16 to 16 (i.e. $[0, 8] \rightarrow [0, 16]$ and $[-8, 0] \rightarrow [-16, 0]$). Turning down the ‘spread’ would contract the pitch range, which may be useful if the range of values were too great (notes too high and too low to be useful to listen too).

The ‘Pitch’ knob can be adjusted from ‘48 to 96 MIDI,’ providing a user-defined value of the transposition of the sonification between C3 and C7. As discussed in Sec. 3.1, this is the MIDI value added to the input data before being converted to cycles per second. The default value is set to the MIDI note 72 or C5. In practice, the ‘Pitch’ knob provides the user with a balance between data

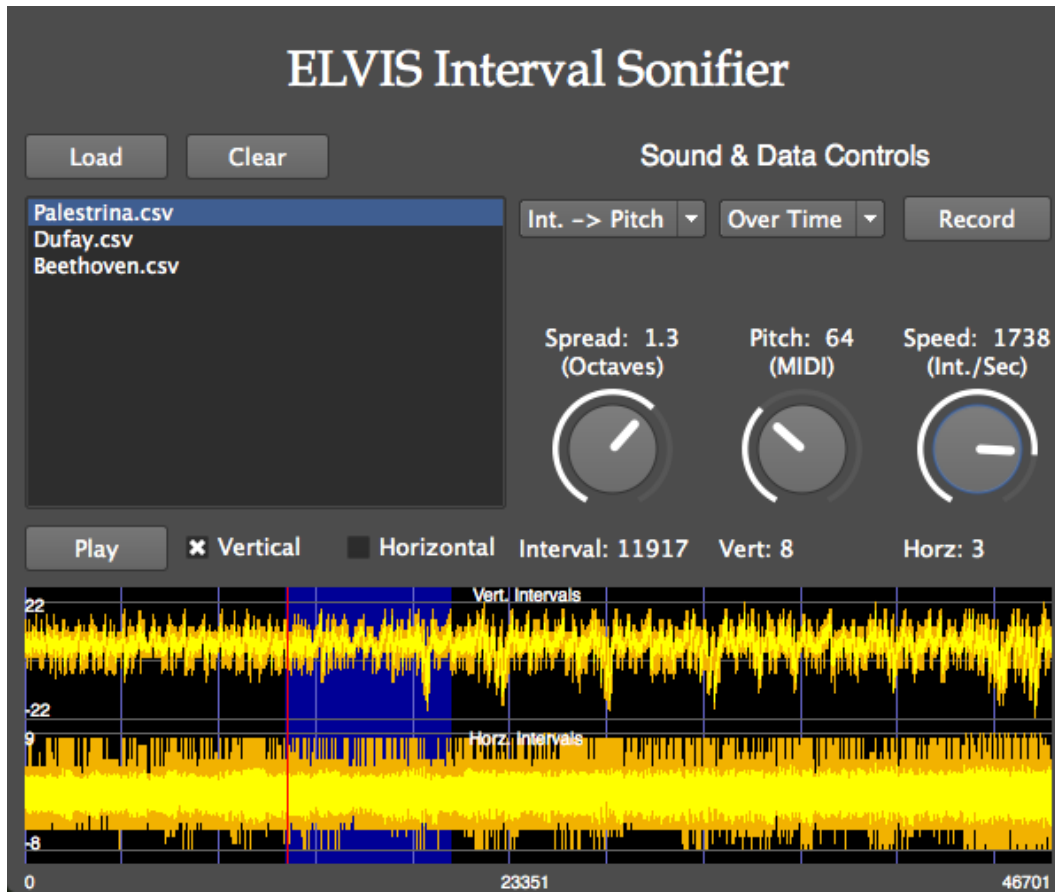


Figure 1: A screenshot of the sonification interface. Data is displayed as a sound-file and the user can make selections and zoom in. The user can choose from available sonification and data controls.

clarity and acoustic comfort. The higher the pitches, the more easily differentiable the pitches become, but also the more annoying or unpleasant to listen to.

The ‘Speed’ knob offers the user the ability to play through the data at speeds from 10^0 to 10^4 intervals per second. Although the technique had previously been applied for exclusively high speed analysis (i.e. 10^2 notes/second [7]), slow playback was made available so that individual musical events might be clearly distinguished—as might be useful if studying one piece rather than a collection. The option of high speed analysis could best be applied when the imported data included thousands of intervals. The interval to interval strategy, while not ideal for high speed analysis (twice as many pitches as necessary), was useful for its ability to represent the imported interval data directly as ascending or descending intervals.

3.3. Data Analysis Features

Although the primary purpose of the interface was sonification, a few basic visual analysis features were implemented. Most clearly, the data being sonified is displayed visually as a sound-file using the QtGUI SoundFileView native to SuperCollider. Below the SoundFileView, the start, middle, and final index of the data in the view is displayed. For both horizontal and vertical intervals, the largest descending and largest ascending intervals are displayed to

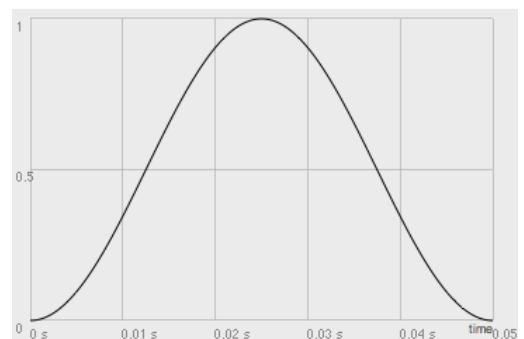


Figure 2: A plot of the amplitude envelope used for sonification generated using the Env.sine envelope generator in SuperCollider. For high-speed playback (> 20 notes/second), each note lasted 50ms.

the left of each track to mark the numerical range of the imported intervals. When the user makes a selection of the data with the mouse, the exact index, vertical and horizontal intervals are displayed directly above the SoundFileView.

As displayed in Fig. 3, data can be displayed and sonified in

one of three ways, ‘Over Time’, ‘Histogram’, or ‘Sorted’. ‘Over Time’ presents the data as it was imported, assumed to be a collection of vertical and horizontal intervals as displayed ‘Over Time’. The ‘Histogram’ option reorganizes the data by collecting the occurrences of each interval type in the data and ordering them from most occurrences to least occurrences. The ‘Sorted’ option simply sorts the intervals in the imported data from largest descending to largest ascending interval. Although both of these options could be realized in other data analysis environments (e.g. Matlab, Excel), offering them in the GUI allowed them to be heard as well—thirds being played concurrently for as long as there were thirds in the data, for example.

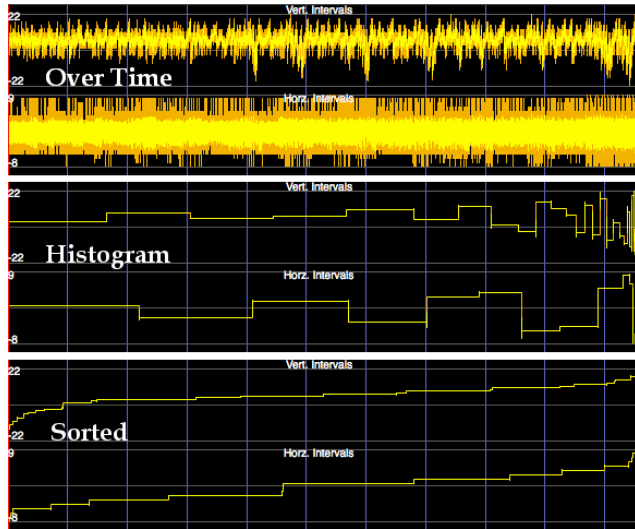


Figure 3: A figure displaying the three data analysis options available for sonification. From top to bottom, horizontal and vertical intervals are displayed ‘Over Time’, as a ‘Histogram’, and ‘Sorted’.

3.4. Distribution

The sonification interface is made available for download in two formats on the elvisproject.ca website.

1. Standalone application for MacOSX 10.6-10.8
2. Supercollider source code

The first option is designed for users unfamiliar with SuperCollider, and works like an application in MacOSX operating systems. When the user boots the application, SuperCollider 3.6.5 runs transparently in the background, handling GUI events, imported data, and running the sonification mapping. The interface can be used on other operating systems, but the user must first install SuperCollider and then run the included source code. The source code will be useful to those wishing to alter or extend the default behaviour, though the standalone application boots the same code from a startup file in the ‘Contents’ Folder. The application also comes with built-in data that is automatically loaded on boot. For users wishing to participate in collaborative development, both versions are available on the ELVIS project GitHub account: github.com/ELVIS-Project.

4. RAPID PITCH MAPPING STUDY

The interface presently described uses a pitch-mapping strategy to rapidly explore large databases of music. Intervals can be presented to the user at high-speeds (up to 10^4 intervals per second) creating characteristic sounds for different composers and genres but also drawing attention to smaller more local structures in the ‘blur’. A study was therefore designed to evaluate the capacity of the strategy to display local ‘events’ that might be of interest in a large amount of data. One such situation might arise when comparing collections of symbolic music that are largely similar. For example, there may be a collection of MIDI files representing a large corpora of music and another collection representing the same music, but arising from a different source, for instance a different performer, or a different algorithm transcribing audio content into its symbolic equivalent.

In this study, pitch-content from sets of two contrived, largely similar corpora are played synchronously through the left and right stereo channels at high speeds using a modified version of the pitch-based mapping strategy introduced in Sec. 3.1. When the pitch is identical in both versions, the corpora are the same, and the pitch is perceived to come from the center of the head. When the pitch is not identical, the two corpora are different and the stream breaks into a pair of two slightly louder, non-identical notes coming from the left and right ear simultaneously.

Instead of using extracted intervals from the ELVIS database, the test uses pitches extracted from built-in corpora of music21, specifically Bach’s Chorales, Beethoven’s String Quartets, and Monteverdi’s Madrigals. For each score in each collection, the .flat method was used to transform every vertical sonority into a horizontal stream with the lowest sounding note played first (e.g., a root position C major chord in four parts becomes the stream {C3, G3, E4, C5}). Each note within the stream was then converted to its MIDI value and appended to a separate list that held all notes extracted from the corpus in order. Using this method, the Monteverdi Madrigals recorded 42,190 notes, Beethoven String Quartets had 167,941 notes, and the Bach Chorales had 125,301 notes. This list was then exported as a CSV file and imported into SuperCollider, which transposed all notes up an octave and a half to increase audibility of low notes.

4.1. Loudness-Compensation Function

In both of the sonification mapping strategies discussed in Sec. 3.1, high playback speeds (e.g. 10^{2-4} intervals/second) generated large amounts of acoustic overlap between adjacent notes and a characteristic increase in global loudness. Further, informal testing revealed that although the spatial divergence cue could be used exclusively, it required an ideal listening environment, slower playback speeds, and concentrated listening. An amplitude compensation function was therefore implemented to equalize loudness for all playback speeds and assist the participant in hearing differences between the corpora.

The equation for amplitude $A(s)$ of each note became

$$A(s) = \frac{1 + \alpha_s \gamma}{1 + \alpha_s}, \quad (1)$$

where α_s is the control of relative gain that varies with sonification speed s , and γ is a gate that is 1 when the corpora are different and 0 when they are the same. Informal testing with the three playback speeds generated values for $\alpha_s = \{60, 15, 4\}$ for $s = \{10^4, 10^3,$

10^2 notes/second. The right level of relative gain gave the impression of an auditory stream that was further away and coming from the center (the similarities), and a second stream that was much closer and coming from the left and right (the differences).

4.2. Generating Differences between Corpora

To create test corpora that were largely the same except for a few differences, each of the three original corpora were copied and randomly chosen pitches were altered using probabilistic pitch distributions. Once modified, the new note replaced the old note in the copy. These two versions were then played through opposing left and right stereo channels.

The probability of note difference between the two versions was fixed to represent the range of probabilities $p(n)$

$$p(n) = \frac{1}{2^n} \text{ where } n \in [0, 1, \dots, 14], \quad (2)$$

where n is an index that is varied to produce a desired probability of a difference. For instance, a given note in the Bach chorales at $p(4)$ had a 1 in 16 ($1/2^4$) chance of being selected for modification.

When a note was chosen for modification using this probabilistic scheme, it was altered from the original by repositioning the note according to gaussian distribution centered around the original and rounded to the nearest integer. The gaussian distribution used in the test was fixed to have a standard deviation of $\sigma^2 = 6$ notes, meaning that the majority of pitch differences spanned less than half of the octave. By perceptual grouping principles [10], this was thought to make the task potentially more difficult for participants than if using larger values of σ^2 .

Pitch differences were created using the method discussed in Equation 2, but for the test, a subset of nine n values were chosen for each of the three sonification speeds:

- For 10^2 notes/second, $n \in [0, 1, \dots, 8]$
- For 10^3 notes/second, $n \in [3, 4, \dots, 11]$
- For 10^4 notes/second, $n \in [6, 7, \dots, 14]$

This method of generating differences resulted in a total of 81 pairs—nine versions for each of the three speeds and three corpora. For testing, a four second sample sonification was randomly chosen from each resulting in ≈ 0 -400, ≈ 0 -500, and ≈ 0 -600 differences for each sound-file at 10^2 , 10^3 , and 10^4 notes/second respectively. Though created probabilistically, for sound-files with low probability of difference (1-10 note differences per sound-file), sound-files were selected to be well ordered, so that the lower probability had approximately half the differences of the next highest probability.

4.3. Methods and Materials

For the test, the 81 sound-files were distributed inside nine folders representing each of the three sonification speeds and corpora. The files within each folder were randomized, but the set of nine folders as a whole was not randomized so that for each participant, Folders 1, 4 and 7 were the chorales, 2, 5 and 8 were the string quartets and 3, 6, and 9 were the madrigals. Likewise, Folders 1, 2 and 3 were 10^2 notes/second, Folders 4, 5 and 6 were 10^3 notes/second, and Folders 7, 8, and 9 were 10^4 notes/second. To better study learning effects, the folders should be randomized for all participants in the future.

Sonifications were recorded as 16-bit AIFF sound-files, and were listened to using Sennheiser HD 800 headphones in the *Witheld for review*. Subjects were instructed to use ‘Finder,’ the default file manager used in MacOS X to preview and order sound-files within the folder. Sound-files were previewed by pressing the spacebar on a standard Apple keyboard, and were dragged and dropped using an Apple Mouse. An example of such a folder containing nine sound-files is shown in Fig. 4 and an ordered folder is shown in Fig. 5.

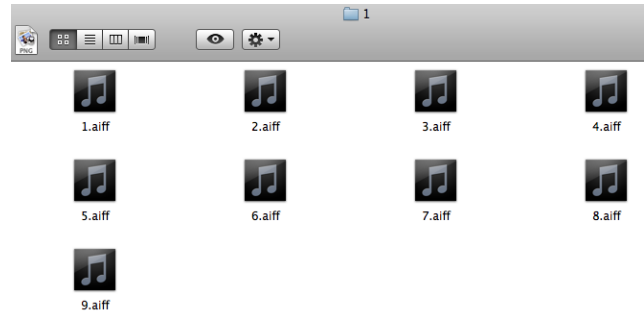


Figure 4: An example folder containing nine unordered four-second sound-files with varying numbers of pitch differences. Participants were asked to order nine of these folders. An example ordering is shown in Fig. 5.



Figure 5: An example of the folder containing the four-second sound-files from Fig. 4 ordered from most note discrepancies to least note discrepancies as determined by the participant.

After explaining to each participant what differences sounded like, the participants were asked to put on the headphones and sound-files from the first folder were played as examples. The participants were also shown how files could be played and paused with the spacebar and how to use the Apple Mouse to arrange files. Once the participant had found an ordering they were happy with, they were instructed to record their answer in written form, which was collected at the end of the test and used for data analysis. With the first folder partially complete, the participants were asked to start with the second folder and complete the first folder after finishing the ninth. This technique allowed the first folder to be used partially for training, and partially as a learning metric—their performance with the last set (Folder 1, 100 notes/sec) being compared to their first set (Folder 2, 100 notes/sec). In future studies, a better strategy would be to isolate a training set from the test folders.

4.4. Participants

The test involved 11 volunteer, unpaid graduate (9) and undergraduate (2) students (4 female, 7 male) studying either music technology (8), information science (1), computer science (1) or psychology (1). All but three had more than 5 years of private music

lessons. Participants were told that the test would last 20–30 minutes and most finished within this time frame. Five participants had heard brief samples when it was demonstrated in a graduate level seminar, and the other participant had heard them several times during development of the technique, been involved with discussions of the technique, and had participated in a pilot test. This participant attained the highest score of any of the participants in the test, but no such enhancement was found for the five whom had heard brief samples.

4.5. Results

A plot of the results from the test is displayed in Fig. 6. The greatest deviation occurs in Folder 2, the first folder that the participants were asked to order. As can be seen, in general participants did very well and parts of folders were ordered perfectly for all participants. The ordering mistakes that did occur tended to be greatest for sound-files with a mid-range of note discrepancies.

Overall, there were very few ordering mistakes made by the participants. Nine out of eleven participants got at least one set perfectly correct. Among this subset of participants, the mean number of sets ordered perfectly was 4.4, the worst performance was three perfect sets ($n = 1$) and the best performance was seven perfect sets ($n = 1$). By speed, the best performance was for the 10^3 notes/second group (Folders 4–6), where the total number of perfect orderings was 19 ($mean = 6.33$) and the highest performance was Folder 4 (Bach Chorales, 10^3 notes/second), which had nine correct orderings.

The high accuracy in the Bach Chorales at 10^3 notes/second (9 perfect orderings) did not continue in the 10^4 notes/second folder (1 perfect ordering). For the three folders at 10^4 notes/second, the total number of perfect orderings was 13 ($mean = 4.33$). The other two folders (8 and 9) at this speed had six correct orderings each which was close to the mean of the 10^3 notes/second group.

The worst performance was in the 10^2 notes/second group which had a total of six perfect orderings ($mean = 2$). Seven participants returned to the first after finishing Folder 9 to complete the partial ordering, and out of them, three ordered it correctly. For the same group of seven, there were no perfect orderings on Folder 2 and two perfect orderings on Folder 3. The number of and type of ordering mistake did not differ between Folders 1 and 3 in this subgroup, indicating that most learning happened in Folder 2.

4.6. Analysis

The results show that the technique was quite effective overall. The best performance was for the 10^3 notes/second group. The 10^2 notes/second group had the worst performance of all three speeds, which may be due in part to learning effects. Two out of the eleven participants did significantly worse than the others, but error analysis revealed that their ordering accuracy was increasing over time.

The difference between corpora was not found to be significant as the effect of speed. Because the loudness cues scale with speed, increasing the value of α_{100} from Equation 1 might result in better performance in the future. Participants found the available cues most useful for categorizing large (> 200) and small (< 50) numbers of pitch differences, and performance tended to be worse for numbers of differences in the middle range.

The balance between localization and loudness cues warrants further study. The loudness cues were incorporated to increase

performance as they could amplify the distinction between correct and incorrectly classified notes. However, in this test, the loudness cues became at times so strong that the spatial cues took a secondary role. Equalizing the loudness between incorrect and correctly classified notes would reveal a threshold for distinction that might be useful to the scientific study of auditory perception.

More information on this study is available in [7].

5. DISCUSSION

5.1. ELVIS Interval Sonifier

The ELVIS sonification interface allows interactive exploration of large collections of intervals extracted from databases of symbolic music. Previously [7], the rapid pitch mapping technique had been applied towards displaying pitch content in large corpora without interactive control. This technique allowed corpora to be distinguished, but without the GUI interface presented in Section 3, it was impossible to probe and view more local structures heard within the dominant ‘blur’. By using the SoundFileView in the GUI, once an interesting sound event is identified, its exact position can be located within the data and listened to at any desired speed. The ‘Spread’ and ‘Pitch’ knobs can adjust the qualities of the sound for clarity and comfort, and also render different sonic ‘views,’ drawing attention to features of the data previously unnoticed.

The finding that listening to extracted pitches of corpora at high speeds could be used to differentiate composers and styles was also found using intervals, though the differences between different composers in similar genres was not as clear as the difference between genres (e.g. romantic quartets sound much different than renaissance masses). Additional findings made possible through the interaction included finding repeated patterns in the data that were not clear in the visual display. Sometimes these patterns were temporally separated from each other, and the fact that they were related may not have been as obvious just by looking. Being able to select when the sonification started by clicking on the SoundFileView and adjusting playback were decisive elements in data exploration.

The interface is also capable of displaying intervals as intervals, which like representing pitch with pitches, is a special use of sonification that may be unique to data-types arising from music or sound. Using sonification this way, data and data representation can sometimes be coupled, referred to as ‘direct mapping’ in Sec. 5.2. Though the benefits of this coupling are difficult to determine, as discussed in Sec. 2.2 they make sound an apt medium for representation, and contributed to the choice of sonification as an analysis tool in the ELVIS project.

5.2. Direct Mapping

Like when sound is used to display audio features [3, 11, 12], sonification of symbolic music creates a special link between data source and sonic representation. Namely, sound is used to represent data that already has a sonic presence. To convey this data, mapping strategies may arise that provide a direct interpretation of the data under investigation.

When browsing large databases of music to find tunes, Fernström and McNamara [13] referred to this type of representation as ‘direct sonification,’ and found that musicologists could use multi-stream audio to complete a musical browsing task faster than

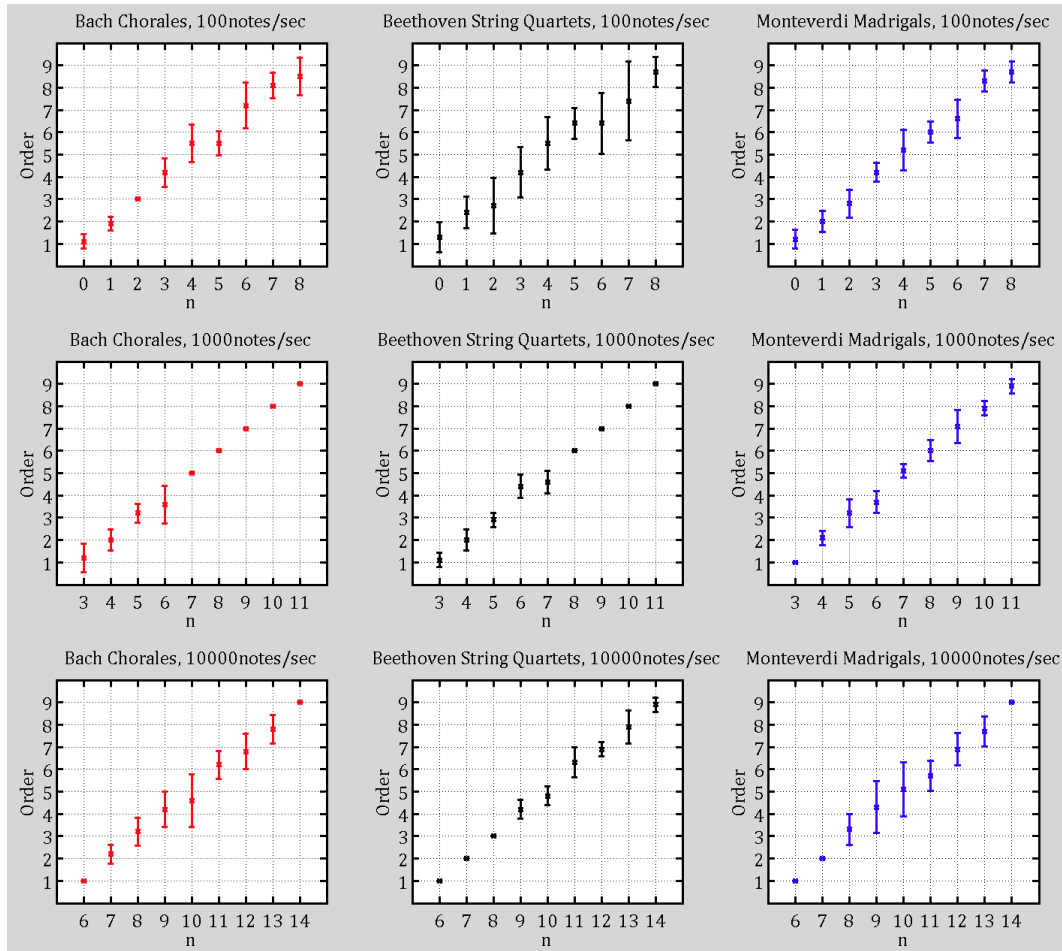


Figure 6: Nine plots showing the performance of all participants on each of the sets. The plots are arranged by number and corpus representing the nine folders the participants were asked to order. The ordinate is the order as arranged by the participants from differences to least differences. The abscissa is the value for n in the probability of pitch difference $p(n)$ in Equation 2. The error bars represent the standard deviation from the mean order number for each n value.

with single-stream audio. More broadly, the interactive browsing of time-based media is sometimes referred to as “scrubbing.” In this paper, sonification mapping strategies were presented in which symbolic musical data was represented by their sonic equivalents. To distinguish the present process from ‘direct sonification’—the transformation does require a synthesis mapping—the term ‘direct mapping’ is introduced.

In the case of symbolic musical data, ‘direct mapping’ occurs when there is an isomorphic transformation of information about sound into its sonic manifestation. It was demonstrated twice in this paper—the interval to interval mapping strategy in ELVIS, and the pitch to pitch mapping strategy in the study. Direct mapping might also arise when representing other symbolic musical elements (e.g. durations, chords, or dynamics). In either case, the information being sonified is not already a sound (i.e. it is not an audio file), but instead a derived symbolic representation. As such, creating sound from this information type is not straightforward, and involves some degree of synthesis mapping.

Though at present difficult to determine contexts in which ‘direct mapping’ might be useful. In the case of music, it was a com-

elling motivation for choosing to apply sonification in the ELVIS project. Combined with the utility of sound for exploring large, complex, and high-dimensional data sets [14], ‘direct mapping’ may be key in the evolution of sonification for this domain.

5.3. Benefits of Sonification in MIR

Outside of this ‘direct mapping,’ this paper has presented three uses of sound for high-speed data exploration and analysis:

1. Differentiating composers, styles, genres
2. Finding non-obvious patterns in the data
3. Comparing differences between similar corpora

Differentiation of composers, styles and genres was discussed in Sec. 5.1. Playing through collections of musical data at high speeds can lead to characteristically different sounds depending on the origin of the musical source. Determining when and why these corpora sound different (or the same) may be useful in directing future analysis. The second benefit, finding non-obvious patterns,

was made possible by the interactive control provided by the sonification interface. Users could visually zoom-in, determine the exact position of interesting events, and manipulate the speed and mapping to uncover what was visually missed. Although corpora as a whole may be differentiable when arising from different composers, styles, and genres, equally interesting are moments when patterns are broken, and sound may be an effective medium for finding these moments. Finally, sound can be used to compare similar corpora, providing a richer cognitive experience than could be provided through strictly quantitative methods.

6. FUTURE WORK

A strategy has not yet been developed for playing through both horizontal and vertical intervals concurrently. This prospect may be useful for finding interesting correlations between the two and might be implemented as two pitches in the left-right stereo channels, or more complexly as a contrapuntal structure between two voices. In the future, better integration with ELVIS software would allow the sonification interface to display metadata about the piece being analyzed for instance, the name of the piece, the parts being analyzed, and the measure and beat of the data point. This metadata would likely be more useful than displaying the index in the imported data, which is provided presently.

7. CONCLUSION

In this paper, sonification was applied to exploring and analyzing intervals and pitches in large corpora of symbolic music. An interactive interface for interval analysis was described offering two mapping strategies, variable playback speed, and controls to maximize acoustic comfort and clarity. Data was displayed as a two-track soundfile representing vertical and horizontal intervals, and when interesting patterns were found through sonification, the user could visually zoom in to locate the exact position of where it occurred. A modified version of the pitch-mapping technique was applied to comparing differences between similar corpora of pitches. A small test demonstrated that the technique could be quickly learned and used to order sonifications by number of differences across three corpora, and three presentation speeds. The possibility for ‘direct mapping’ was determined to be a special quality to contexts of sonifying data from music, though other benefits of using sound including differentiating corpora and identifying non-obvious patterns were also highlighted.

When exploring large databases of music or information derived from music, sound offers a unique medium for data display that can at times transcend data and representation. Sound can provide a rich cognitive experience of musical data and has usefulness as a analysis medium as well. Future applications of sonification in this context may help transition the use of sound from a tacit medium for displaying final results to integrated tool in musical discovery.

8. ACKNOWLEDGEMENTS

Funding for this project was made possible by the Social Sciences and Humanities Research Council of Canada (SSHRC), Digging into Data Challenge Grant, 2012-2013. “ELVIS: Electronic Locator of Vertical Interval Successions. The first large data-driven

research project on musical style.” Julie E. Cumming, PI. The authors would like to acknowledge the helpful advice from anonymous reviewers, including the reference to ‘Direct Sonification’ in [13]. The weekly contributions of musicologists, theorists, historians, and computer scientists in the ELVIS team were decisive in bringing the sonification interface to its completed form.

9. REFERENCES

- [1] P. Vickers and B. Hogg, “Sonification abstraite/sonification concrète: An ‘aesthetic perspective space’ for classifying auditory displays in the ars musica domain,” in *Proceedings of the 12th International Conference on Auditory Display*, London, UK, June 2006, pp. 210–6.
- [2] T. Hermann, “Taxonomy and definitions for sonification and auditory display,” in *Proceedings of the 14th International Conference on Auditory Display*, Paris, France, June 2008.
- [3] S. Ferguson and D. Cabrera, “Auditory spectral summarisation for audio signals with musical applications,” in *Proceedings of the 10th International Society for Music Information Retrieval Conference*, Kobe, Japan, 2009, pp. 567–72.
- [4] S. Ewert, M. Müller, and P. Grosche, “High resolution audio synchronization using chroma onset features,” in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Taipei, Taiwan, 2009, pp. 1869–72.
- [5] C. Antila and J. E. Cumming, “The VIS framework: Analyzing counterpoint in large datasets,” October 2014, submitted.
- [6] M. S. Cuthbert and C. Ariza, “music21: A toolkit for computer-aided musicology and symbolic music data,” in *Proceedings of the 11th International Society for Music Information Retrieval Conference*, Utrecht, Netherlands, 2010, pp. 637–42.
- [7] R. M. Winters, “Exploring music through sound: Sonification of emotion, gesture, and corpora,” Chp. 7, McGill University, Montréal, Canada, August 2013.
- [8] S. Wilson, D. Cottle, and N. Collins, Eds., *The SuperCollider Book*. Cambridge, MA: MIT Press, 2011.
- [9] M. Epstein and J. Marozeau, “Loudness and intensity coding,” in *The Oxford Handbook of Auditory Science: Hearing*, C. Plack, Ed. New York, NY: Oxford University Press, 2010, vol. 3, ch. 3.
- [10] A. Bergman, *Auditory Scene Analysis*. Cambridge, MA: MIT Press, 1990.
- [11] D. Cabrera and S. Ferguson, “Sonification of sound: Tools for teaching acoustics and audio,” in *Proceedings of the 13th International Conference on Auditory Display*, Montréal, Canada, 2007, pp. 483–90.
- [12] S. Ferguson, “Statistical sonifications for the investigation of sound,” Ph.D. dissertation, University of Sydney, Sydney, Australia, 2009.
- [13] M. Fernström and C. McNamara, “After direct manipulation—direct sonification,” *ACM Transactions on Applied Perception*, vol. 2, no. 4, pp. 495–9, 2005.
- [14] B. N. Walker and M. A. Nees, “Theory of sonification,” in *The Sonification Handbook*, T. Hermann, A. Hunt, and J. G. Neuhoff, Eds. Berlin, Germany: Logos Verlag, 2011, ch. 1, pp. 9–39.