Aesthetic Sonification Toolkit for Real-time Interaction with Data

Professor Kirsty Beilharz
University of Technology Sydney
P.O. Box 123 Broadway 2007 Australia
kirsty.beilharz@uts.edu.au

Dr. Samuel Ferguson
University of Sydney
Wilkinson Bldg G04 2006 Australia
samuel.john.ferguson@gmail.com

ABSTRACT

This paper presents a new approach to aesthetic sonification framework to enable user-interface customisation, aesthetic control and interaction with datasets. In particular it provides tools for real-time interaction with data in a number of configurations of pitch, modality, timbre, focusing on ergonomics of listening, aesthetics of sonification, and techniques of sound production for data analysis. Firstly, we look at the role of the system and ways of thinking about information; secondly, we review existing sonification frameworks to position our approach; and finally, we explain our method.

Keywords

Sonification, Auditory Display, Aesthetics, Data Analysis.

1. INTRODUCTION

1.1 Sonification and Auditory Display

Auditory Display is a broad term referring to any use of sound to provide information to a listener. Walker and Kramer define various categories of auditory presentation within this broad field; *Alerts and notifications* are the simplest type of auditory display. *Auditory icons* are the auditory equivalent of the visual icon (eg. rainy weather could be represented by the sound of rain on a roof). *Earcons* are sounds in an interface that do not have such a direct relationship to their object, but instead substitute a hierarchical musical language, and result in the necessity to learn the meanings of each earcon. *Audification* is the transformation of a fluctuating data source (like a seismogram) directly to an audio waveform. *Process Monitoring* refers to displays that represent processes that allow the user to monitor the multiple streams of data in a manner that would be difficult to do with visual displays. And finally, *Sonification* is the use of non-speech audio to convey information such as that used in the interpretation of scientific results [1].

Of all of these categories, Sonification is the only one that requires a software framework for it to occur. All the other categories are generally sounds that are designed and then incorporated into a larger project. Audification is the exception, but this is still a straightforward process, simply requiring data normalisation, time domain synchronisation and conversion to audio format. The possible methods for sonification, however, are not bounded in such specific manners. Indeed, the sonification software employed often forms the boundaries.

1.2 Analytical Representation Purposes

Sonification is the use of non-speech audio to convey information, and therefore general principles of analytic thinking and analytic design are essential to its understanding. Tufte argues that analytical design principles are universal and generic, and stem from analytic thinking principles [2]. He outlines some principles of analytic thinking:

Show Comparison, Contrast, Difference: the information regarding change is usually found through a difference being detected. *Show Causality, Mechanism:* Simply presenting the contrast or difference does not often describe the cause, which is essential to understanding. *Multivariate Data:* By presenting more than one or two streams of data, many more inferences can be drawn. *Integrate Words, Numbers, Diagrams:* These assist in providing context and information regarding the important parts of the data. *Thoroughly Describe the Evidence:* Representations should include a title, describe the sources of the data and sponsors, use scales and point out other issues. To support these rules, Tufte uses the multivariate graphic of Napoleon's march into Russia and subsequent retreat by Minard.

1.3 Computational Information Design

Similarly, Fry has described the synthesis of fields such as data mining, statistics, graphic design and information visualization into a practice he describes as 'Computational Information Design' [3]. It is a design process that involves 7 steps. A simple data representation example (from Fry) may be described using these steps as – data about the postcode system is *acquired* from a website, *parsed* into fields representing latitude and longitude, *filtered* for a particular region's postcodes, *mined* to work out how large a representation grid should be, *represented* as points on a longitude/latitude grid, *refined* by altering contrast and colour attributes, and *interacted* with by using zoom and label actions. Similar processes can be design patterns that occur in sonification.

1.4 Aesthetic Sonification Toolkit Aim

Firstly, we will summarise the aims of the aesthetic sonification toolkit design presented here. The following section will review some of the existing sonification toolkits and frameworks, to illustrate the distinctions and to delineate why we believe it is necessary to develop another, explaining how ours is distinctive in its functionality and goals and how it may be applied pervasively to a versatile array of datasets.

Visualization is widely becoming recognised both as a functional, communicative way to convey information and as a medium that can be aesthetic, persuasive and attractive in its delivery. Visualization methods encode datasets through graphic representation, while sonification uses a variety of auditory encodings to produce an auditory representation. Tufte [4], Fry [3] and others have argued that visualization can learn from principles of good graphic design – notions such as colour hierarchy, affects of scaling, semantics of certain signifiers, contextual relevance, the necessity to differentiate competing streams of information occurring in the same space – perhaps sonification may equally learn from music and good sound design. One of the problems that can arise from automated practices of transforming data streams into digital audio signals is that often the computational representative process can dominate, leading to a limited scope of auditory dimensions which eventually produce a boring, unattractive or even confounding result. Above all, we view it as extremely important to convey information in an efficient and aesthetic way if it is to be ergonomic, i.e. listenable over a period of time, and not too repetitious or annoying.

Naturally, aesthetics is subjective and individual, so there are two approaches we employ to tackle this problem. At a gross level, one can generalise about ideas like consonance vs. dissonance, tonality or modality, avoiding harmonic and pitch clashes for reasons of distaste and confusion of messages. One can also generalise, based on musical and sound design experience, about the use of spatialisation and timbre (tone colour / tonal quality) to distinguish between different streams of information heard at the same time to aid concurrent deciphering. Furthermore, background in auditory cognition and perception informs decisions about the optimal range for audibility and the contextual and orthogonal influences of loudness, approaches, to avoid temporal and spectral masking [5, 6]. Therefore, there are some established listening guidelines informed by psychology and music that can already frame parameter choices and their scope for sonification. These have been carefully investigated by authors such as Walker [7] and Neuhoff [8] among others as to their applicability in sonification.

However, at the individual, subtle level, the way in which we aim to make our sonification toolkit aesthetic is to open several parameters of each dimension to user-customisation and contextual scaling. This means that the user can adapt the scale/scope to suit the data in a preparatory grooming stage and then, using real-time controls, adjust the playback experience according to preference. Interaction can also tap into a key objective of sonification – that is to realise more interesting and revealing features of the dataset. Sometimes, playing an episode repeatedly but using different approaches such as changing direction, faster or slower tempo, removing single features, highlighting or emphasising certain streams can elucidate trends, anomalies and patterns in the data that are otherwise more difficult to discover.

Other reasons for aesthetic sonification include the diversifying contexts for auditory and visual display enabled by pervasive computing. Increasingly, we are seeing the integration of auditory and visual or bi-modal information display systems in wearable computing (e-fashion, smart-wear), e-jewellery or other body-mounted information displays representing personal and public data about the user. Such information may include abstract rendering of bio-data, feedback from bio-, motion- and geographical sensors worn by individuals for an array of functions from medical, fashion to socially interactive and provocative. Information is graphed and displayed increasingly in our environments, such as eco-aware buildings reporting on their sustainability or occupant behaviour, channelling surveillance and sensor data, building intelligence into socially meaningful reflections of activity, resource usage, greenhouse and energy efficiency, and so on. Many ubiquitous computing devices like handheld PDAS, mobile phones and in-car devices give us regional, geographical, topographical, climate, traffic and other relevant reporting on mobile contexts. Driving a car, for example, mobile phone usage is linked to distraction and incidents of accidents or near-miss scenarios. Devices that can convey useful information without averting our eyes from a task at hand serve a very useful purpose, established already in work environments such as air traffic control, air-pilot cockpits and for peripheral and ambient observation of information. For reasons of ergonomics, aesthetic listenability and variability are useful controls for sustained workplace information monitoring. For reasons of individuality, expression and choice, aesthetic individualisation and customisation are essential to e-fashion and the living and working environments we inhabit. Sonification can often co-exist in a bi-modal display or support and reinforce a temporal, ambient visualization in such everyday contexts but we all know the vital necessity of appealing, changeable and meaningful sounds in contrast to the rapid annoyance and irritation provoked by un-aesthetic alerts, alarms and monitoring devices, of which sonification is perhaps a more protracted and continuous example.

One way in which we attempt to augment the aesthetic scope of the sonification is to employ some auditory dimensions typically given low priority. Most sonification mapping schemes focus on graphing linear temporal spacing to a form of meter or periodicity. However, time can be used to represent time-series data – periodicity can naturally encode feelings of urgency, concurrence or stasis. Hence, we also use rhythmic pulses to represent facets of the data, but allow flexibility of tempo to reveal trends and points at different tempos. The majority of sonifications use frequency or pitch on one axis of the data graph. While we also believe pitch is an important criteria, we look to alternative modalities and scaling or spacing to again elucidate different levels of attention or minutiae and adjustable smoothing that optimises or highlights key data points. Changing mode and harmony is also an easy way to eliminate boredom and repetitiveness. In addition, we give considerable attention to some parameters not widely utilised that can offer helpful benefits in combination with other dimensions, such as spatialisation, reverberation (producing a spatial dimensionality for distributing information in virtual auditory space) and timbre (tone colour) to reinforce the message of other dimensions and highlight events occurring beyond certain thresholds and bring to the foreground events of informational significance.

In summary, the priorities of this Aesthetic Sonification Toolkit are:

- aesthetics
- interactivity
- adjustability/customisability
- applicability to simple time-series and multivariate datasets
- real-time data processing

1.5 Sonification Frameworks

In designing our toolkit we reviewed other sonification software frameworks as reference points that describe the methods and systems for creating sonifications.

1.5.1 Listen, Muse, Musart

Some of the earliest frameworks, the Listen, Muse and MUSART frameworks overseen by Lodha [9-11] are research sonification platforms that developed many of the basic patterns for sonification frameworks. 'Listen' mapped data to parameters of MIDI synthesis – pitch, duration, volume, and location. 'Muse' extended this model and added many more musical elements as mapping options – it used CSound for synthesis, and its purpose was to produce sonifications of a more musical nature. Finally MUSART extends this idea using 'audio transfer functions', and adds still more mapping possibilities. They are notable for their early experimentations – and a lot of the ideas they put forward have been incorporated into later frameworks, such as the Sonification Sandbox.

1.5.2 Sonification Sandbox

http://sonify.psych.gatech.edu/research/auditorygraphs/sandbox.php

The Sonification Sandbox is a project of the School of Psychology at the Georgia Institute of Technology, overseen by Bruce N. Walker [12, 13]. It is motivated to provide a multi-platform, multi-purpose sonification toolkit. It enables the user to map datasets to any dimensions from timbre, pitch, volume and pan. The toolkit also provides some tools for modifying the way the data is analysed, for example examining mean, maximum, minimum or providing notification in relation to the data mean and direction changes or trends. It works with universal file formats such as Microsoft's Excel, .CSV and exports MIDI or QuickTime play files.

The Sonification Sandbox is a mature framework that allows a particular type of auditory graph to be produced with the minimum of effort. It works with tabular datasets, and produces both visual graphs and auditory graphs, in a similar way to Excel. Vickers has discussed the Sonification Sandbox, and has pointed out some of the problems inherent in the reliance upon the MIDI protocol [14], and indeed Davison and Walker do as well [13] – however they argue that the sandbox is not meant to produce every possible sonification design, but rather is positioned as a simple general-purpose tool to build auditory graphs. It remains a well-maintained piece of software, and many of the options for sonification it provides have been supported by the author's research (for instance Walker [7]).

1.5.3 SonEnvir

http://sonenvir.at/

Developed by four Universities in Graz, Austria (Karl Franzens University, The University of Technology, the Medical University, and the University for Music and Dramatic Arts), SonEnvir is built on SuperCollider synthesizer/programming environment. It specifically targets the demands of fields with complex, multi-dimensional data for analysis, hence the affiliation of its collaborators. One of its drawbacks is the significant amount of knowledge required to use its implementation in SuperCollider. It is quite technically ambitious, and also seems specifically designed for the IEM Cube spatial playback system. To modify the template requires significant programming understanding, and it seems that such sonification is often undertaken in a collaborative research context, such as in the 'Science by Ear' Workshop [15]. It benefits from a wide user-base, and a well-organised website and is interesting in the diversity of data that it has been applied to [16].

1.5.4 SoniPy

http://www.avatar.com.au/sonipy/

David Worrall's *SoniPy*, as the name suggests, uses the Python programming language as its foundation [17, 18]. He characterises the development purpose as a need to balance the data processing and sonification capabilities within one piece of software. His premise is that others will embrace the development of new modules and add-ins using this modular programming platform. However, he acknowledges the quality of the modules available can vary significantly, and the installation of each of these tools can increase the complexity of the process. At this stage no downloads seem available from the repository.

1.5.5 Monalisa

http://www.monalisa-au.org/

Monalisa by Norihisa Nagano (IAMAS) and Kazuhiro Jo (u-Tokyo RCAST), is a sonification toolkit that transforms image filtering and textual information (respectively) into data sonification of files on your computer's hard drive. While a serious sonification tool, its purpose might also be construed as exploratory and investigative or curious and fun, more than for information analysis. It reinforces correlation between visual images and auditory timbre by an immediate "translation" or "transliteration" of visual into auditory images. It operates in a few different modes: in the *Monalisa* Application, it enables the user to "see the sound, hear the image"; it also has an Audio Unit and Image Unit mode (whereby audio effects are applied as image effects or vice versa), as well as an incarnation as an installation performed with video. It operates by converting image data into audio data or conversely audio data into image representation. At the conversion stage, the image or audio filters are applied, altering the resulting audio or image in a fairly unpredictable manner. A single parse of the file data produces a single rendering as sound or vision (depending on the direction in which it is applied). While its outcomes are quite diverse and feature a range of aesthetic results, its intention is not to provide a tool for analysing and rendering datasets – it is specifically interested in the crossover between sound and image.

1.5.6 SonART

http://www-ccrma.stanford.edu/~woony/software/sonart/

SonART is a flexible framework for real-time sonification implemented using Java and OSC by Woon Seung Yeo, Joanthan Berger and R.Scott Wilson at the Centre for Computer Research in Music and Acoustics, Stanford University [19, 20]. The OSC network communication facilitates a variety of real-time synthesis and distributed synthesis options; currently implemented as a Cocoa-based OSX application. The current version builds on a mapping framework described by Ben-Tal et al. in 2002 [21]. Their stated aims focus on network transmission of data, real-time sound generation, distributed synthesis and cross platform, and modular design. This part is intended for multi-user (network shared) collaborations or data sharing and conceivably mixing data from more than one locality source. In recent incarnations the emphasis seems to have shifted towards more performative and graphic-centred sonification, using the image data and layering effects to generate new images and sound. The data sonification is specifically from images or data linked to images. In this regard, it is closer to the *Monalisa* approach, both parsing image data for sonification. Its developers view SonART's potential industry application in connection to image sonification and analysis (such as for medical imaging diagnostics, security and cell structures [in imaging]).

1.5.7 Interactive Sonification Toolkit

Sandra Pauletto and Andy Hunt's Interactive sonification Toolkit was developed in the Media Engineering Group at the University of York. They describe an environment in which datasets are scaled and then may be sonified in a variety of ways and navigated with the mouse (with future plans for other interface controllers) [22]. Their toolkit principally deals with non-real-time datasets of pre-gathered information (although an on-the-fly theory is posited) and the aim lies in applying various sonification algorithmic processes to the same dataset to produce various auditory outcomes. It is cross-platform and uses Pure Data (PD) as its programming foundation. Part of the interaction involves the user choosing the maximum and minimum data points and scaling and stretching factors. A separate and subsequent process converts the data to sonic representation. They propose a number of different ways of listening to sounds, ranging from distinct data dimensions mapped onto distinct auditory dimensions, to generating a complex timbral effect whose single-sound complexity reflects data structure. They also explore a number of different methods for data mapping, like note duration mapping, audification, pitch mapping, noise filtering and additive synthesis. The system seems designed around a static user interface that presents the sonification options that are currently built into the system, than using the modularity that some of the other frameworks seem to favour. Pauletto and Hunt's is intended to be a generic system applicable to a range of data types.

1.5.8 Personify

Barrass discusses *Personify* in the context of a number of sonifications built using the TaDa design template [23], the topic of his Ph.D. The tool is divided into two parts. The requirements part sets up the data according to the TaDa method, as well as configuring a default representation. Histograms are computed and mapping extremes can be configured, as well as data types. The representation part has a number of methods for representation and a number of output audio devices.

2. IMPLEMENTATION DESIGN

In planning the implementation of our framework, there are a few tradeoffs that can be described regarding the aforementioned toolkits.

Flexibility vs. Usability: Outstanding flexibility of a sonification framework is usually afforded by the use of modular object oriented languages, such as Python or SuperCollider, and by providing a set of interlocking 'classes' to develop sonification programs with. Other systems (Musart, Sonification Sandbox) tend to provide particular user interfaces associated with controlling particular sonification styles.

MIDI vs. Audio Synthesis: Some of the simplest mapping possibilities are afforded by those systems that use MIDI (Sonification Sandbox, LISTEN); by restricting themselves to the MIDI protocol they avoid the confusing array of options provided by current digital audio synthesis whilst achieving a perceptually relevant range of audio parameters for representation. Those frameworks that are designed around audio synthesis are commonly more complex, and require more knowledge of audio synthesis techniques to explore them.

Open Source vs. Closed Source, Platform Dependency: Many of the sonification frameworks reviewed see open-source release of their software as useful for future research. Many are also built using cross-platform techniques – sadly some that haven't have been eclipsed by their target platform's demise (e.g. Listen, MUSE, Musart – all built on the SGI platform). Only a few frameworks seem to have active download pages available, however, and a number of these are source control repositories that require specific client software to connect to them.

Aesthetics – an extra level of complexity. Many of the simpler frameworks provide few controls over sound quality. Few think about aspects such as chordal momentum or synthesis aesthetics as the user interface complexities are immense – when dealing with unknown datasets, it seems a common approach is to have a data parsing stage, a mapping stage, and an output stage. A further imposition of a set of musical and aesthetic priorities has seemingly not been attempted, except in environments such as *SonEnvir*, where the entire sonification process is developed using SuperCollider classes. We aim to provide more formalised tools to deal with the interaction between representation mapping and aesthetic concerns, hopefully developing patterns of design that can be exploited.

2.1 Platform: Max/MSP and FTM

Max/MSP is a flexible software platform for prototyping, and development of audio/visual programs. However, the data structures and mapping options available in Max/MSP are generally quite limited, especially for the statistical type of data structures necessary for sonification purposes. FTM is a shared library add-on to Max/MSP that provides flexible data structures, as well as commands for dealing with matrix calculations, thus avoiding most of the main limitations of Max/MSP, while still retaining much of the usefulness of the flow-based programming metaphor [24-26]. It provides data structures extending from matrices of floating point numbers to full SQL database functionality, significantly increasing the ease with which data can be manipulated.

The most useful aspect of Max/MSP is the flow-based programming metaphor. This is a common method for developing audio programs for other purposes, and allows a huge amount of flexibility, while avoiding the specificity of text-based languages. Not all categories of programs are easily implemented using the flow-based paradigm, and modern programming often use object-oriented techniques for various reasons, however, musicians and sound designers are often more comfortable with a flow-based metaphor due to similar flow-based experiences using music studios. Users typically find the metaphors in flow-based programming simpler to understand than they do complex ideas used in text-based object-oriented languages, and this is one reason we chose this language. Furthermore, flow-based programming in this environment maintains little distinction between the user interface to the program, and the program itself, interactive updates of mappings and experimentation between various options are all possible while the program is running. Other languages may require the relationship between the mapping and the user interaction to be defined as a user interface control before the program is recompiled and re-executed.

3. OBJECT CATEGORIES

This framework exists as a set of Max/MSP *objects*, the building blocks used in Max/MSP. The framework is built around a core of these objects, and user interface elements are 'bolted on' last. There are five types of category for the objects in this framework. *Dataset* objects deal with acquisition, parsing and annotation of data. *Mapping* objects transform data to other types of data, but do not create sound – they do not map data to sound in one step. *Synthesis* objects create sound, and usually output audio rate signals. *Manipulate* objects use data to alter an audio rate signal in some manner. *Interface* objects perform interface tasks, such as taking in user input and displaying visual feedback to assist users to choose configuration parameters (see Figure 1). A typical sonification program would involve *Dataset* category objects, connected to *Mapping* category objects, connected to *Synthesis* category objects, connected to *Manipulate* category objects, finally connected to built-in audio output objects. *Interface* category objects may be attached to any of the other four category objects to allow user interaction.

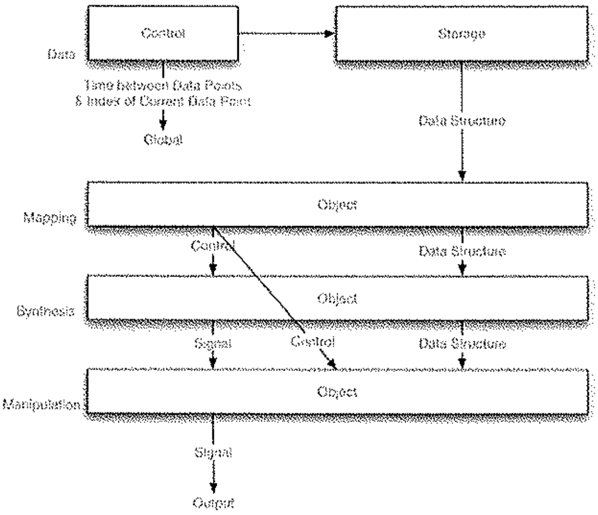


Figure 1. Relationship between object categories in the Sonification Framework.

3.1 Dataset objects

One of the benefits of using FTM as an extension to Max/MSP is that there are built-in data structures that are much more appropriate for dealing with large, complex datasets. They also include matrix calculation commands that are very helpful in a flow-based language such as Max/MSP. Therefore the dataset is dealt with as a set of FTM data structures, which are passed as references to the remainder of the objects. These data structures are passed to each object, and commands within each object select the appropriate element from the data structure, perform the mapping, and output the results. There are other dataset objects that attempt to support the main data structures, through summarising the size and shape of the data, and providing control of the manner in which the data is stepped through.

The use of FTM also means that data is passed by reference; by altering the original data, all the objects that are associated with that data are updated automatically. The benefit is that this allows real-time data sources and static data sources to be dealt with in very similar ways, and to be swapped at will.

3.2 Mapping objects

Mapping objects take input data from the Dataset objects and output control data. Essentially they contain a mathematical equation, which takes in data points of one range and shape and maps it to another range and shape (eg. normalisation, logarithmic transformation, deviation calculation, or quantisation). They may affect the entire data shape, using its attributes as inputs to the mapping algorithm, or may function on only the currently rendering data point. Their output is either control data (such as triggers for manipulate objects or pitches for synthesis objects) or altered data structures to be sent on to further objects.

3.3 Synthesis objects

As this framework is designed to be flexible, the synthesis options are not strictly defined. Therefore, the outputs of the mapping objects as easily used in typical MSP synthesis methods. Simple synthesis techniques, such as frequency modulation, additive or subtractive synthesis are included as *Synthesis* objects to work with the outputs of the mapping objects directly. Networks of these objects can be built to create more complex auditory possibilities. These objects do not require any input from the dataset – control data created at the mapping stage is sufficient.

3.4 Manipulate objects

Manipulate objects map data to an alteration of an existing audio signal, produced by a synthesis object of some type. The simplest example is that of a gain control – it may amplify or attenuate a signal according to incoming data values. Other methods that fall into this category include filters, envelope functions, compressors and reverberation algorithms. At the more obscure end of the possibilities for these objects we can find granulation and time-stretching objects.

3.5 Interface objects

Interface objects provide ways of accepting user interaction information to control aspects of the sonification, and providing visual feedback. They have outputs that can be again scaled and mapped to be inputs that are used by other objects in the framework.

An example of an interface object is the Wii transport control object. The Nintendo Wii remote is a well-known sensor device that contains a number of accelerometers. Using an internal inertia calculation the tilt of the device can be used to control the speed at which the data points are presented, utilising a balance metaphor.

4. SONIFICATION OPTIONS

The new methods we define in this sonification toolkit are presented below, and their basis in musical aesthetics is discussed.

4.1 Timeline

The time axis is a fundamental auditory parameter - without it no sound could exist, and no auditory representations could be varied. Usually it is used in a completely linear, flat manner, with data points presented one after the other until the dataset is exhausted. This is in contrast to typical musical forms, which have several levels of time division. It could be seen that these levels are useful in providing context and rhythmical vitality to the auditory presentation, and certainly they appear in most western musical styles, even extending over into other art forms such as poetry.

The rhythmical levels could usefully divide time at levels analogous to the *beat*, the *bar*, and the *stanza*. The fundamental time unit is the *beat*, which is configured in milliseconds. In most situations a single data point will be presented per beat. The *bar* is a multiple of the beat, and can be used to group data points. This may be arbitrary, but in most instances there are data attributes that can be used to find a multiple that is appropriate. For instance, a dataset of monthly totals over a number of years can be presented with a 'bar' of 6 or 12 to group the data into half-yearly or yearly groups. The *stanza* is a less important division that is configured as a number of bars.

Of course, the timeline is used to determine the rate at which the data points are presented. However, it is also used to trigger short rhythmical samples at each *bar*, to trigger longer sound-bed samples at each *stanza*, and to trigger changes in parameter mappings at particular points in the progression of the sonification.

4.2 Pitch

Pitch mapping commonly occurs as a simple mapping to exponential frequency. Linear data is transformed to frequency values that are then synthesised directly. The use of a 12-tone per octave logarithmic scale (a chromatic scale) for representation of numerical data is common and simple, but perhaps not an ideal choice in terms of aesthetic considerations. If two sets of data are played at the same time, it is impossible to control the harmonic interrelations between the two, and arbitrary dissonance or consonance may occur. If the aesthetics of the sonification are important, more options for quantisation of pitches to particular scales are necessary.

Our method defines two sets of parameters for the mapping. The output pitch range to be mapped to is defined, and is associated to statistical aspects of the data. The most convenient method is to associate the maximum and minimum to the maximum and minimum of the output pitch range, although for other situations it may be useful to use other statistical characteristics of the data, such as the inter-quartile range, or the 1st and 99th percentiles. The default we use in this case is to associate the range of the data to the output pitch range.

The input data is transformed and then quantised to the set of pitches we define. The pitch quantisation set is defined in relation to the octave, so that the set can be extended over large ranges as well as small. This pitch quantisation set can be set to be the chromatic scale, or perhaps major or minor scales. Various scales have various characteristics of course, based in the inherent relationships that pitch perception exhibits. Pitch quantisation sets such as the chromatic scale, augmented scale or diminished scale have no clear octave boundary, whilst major, minor and pentatonic scales, and choices of notes that form chords, do include this boundary. In some situations it may be attractive to use a pitch quantisation set such as the augmented scale, to achieve a sonification without clear octave boundaries. Furthermore, the pitch quantisation set is not limited to equal tempered (MIDI note) numbers, and fractional values may be used as well.

Finally, the quantisation set may be changed at any point throughout the sonification. Initially, this may seem irrelevant and unnecessary. However, when the data points are presented rapidly, and the quantisation set are pitches that form a triad of some type, the result is likely to be perceived as a chord. If the quantisation set is changed to notes that are within a similar pitch range, but are a triad from a different chord, then harmonic momentum may be built. These quantisation set changes may be attached to particular points in the timeline, or they may be used as mappings of a categorical nature, denoting a different category of data is being listened to.

4.3 Harmony

Harmony has not often been investigated for use within data sonification. Sonification generally maps data to attributes of single streams of sound, and therefore the interaction between the pitch attribute of two independent streams – harmony – is a difficult target for mapping. Rather than using pitch as the mapping for two separate streams, if it is used only for one, and the second stream is based on a harmony being added to the original pitch, the harmony may be used for a categorical marker. Simple quantisation sets of harmonies, such as three intervals, are likely to be more successful aesthetically. Using several iterations of design it is possible to build harmonic mappings that form unexpected relationships or meanings. In a similar manner to pitch, harmony quantisation sets may be changed dynamically, leading to harmonic momentum, and building musical forms.

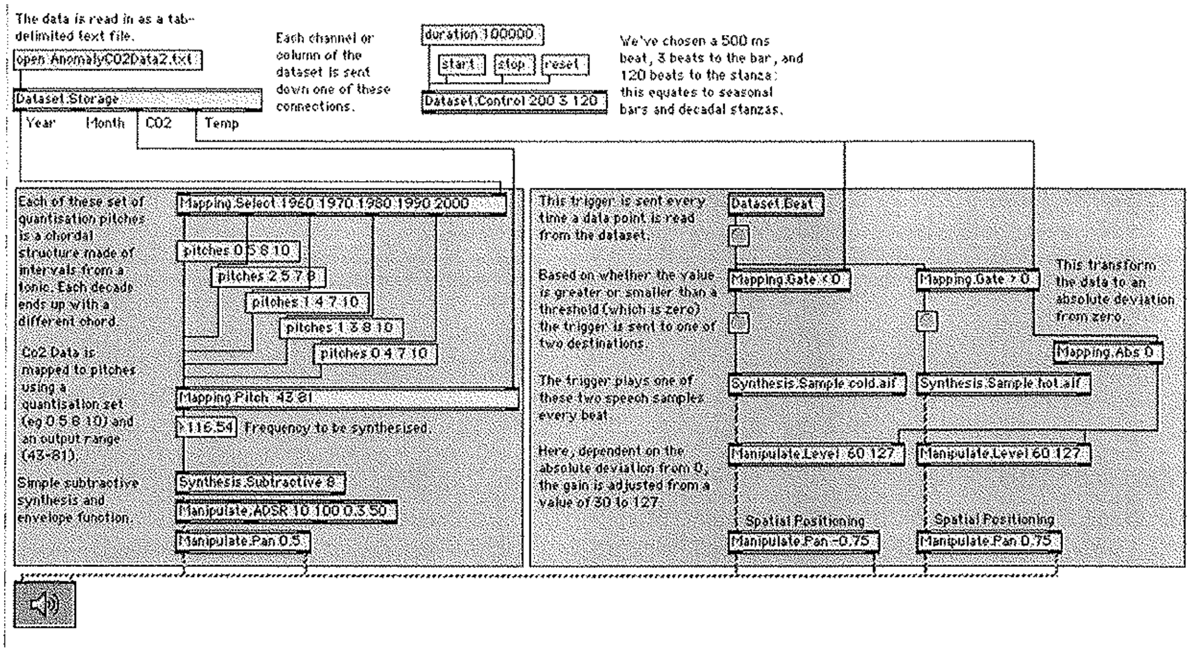
4.4 Rhythm

Adding rhythmical interest to sonification helps to build diversity into the sonification result. Some methods for achieving rhythmic interest include offsetting separate streams, rhythmically dividing particular notes to be repeated and sounding the bar lines. 'Muse' [10] defined several rhythmical levels based on dance styles, where emphases were altered. At this stage we use simple divisions of the beat into 2s, 3s and further multiples of the beat.

Emphases can also be controlled by a *Manipulate* category object that uses bar lengths, beat lengths and gain values to create a gain function that can be applied to a signal.

4.5 Spatiality

Single (mono) auditory images are simple, but probably not ideal for presenting multivariate information. Of course, spatial separation is very important for aesthetic purposes, with much research investigating spatial presentation in audio reproduction. However, Song et al. also finds that separating two streams of sonified data increases their comprehensibility [27]. Furthermore, sonification projects with aesthetic



The data is read in as a tab-delimited text file.

Each channel or column of the dataset is sent down one of these connections.

We've chosen a 500 ms beat, 3 beats to the bar, and 120 beats to the stanza: this equates to seasonal bars and decadal stanzas.

Each of these set of quantisation pitches is a chordal structure made of intervals from a tonic. Each decade ends up with a different chord.

Co2 Data is mapped to pitches using a quantisation set (eg 0 5 8 10) and an output range (43-81).

Simple subtractive synthesis and envelope function.

This trigger is sent every time a data point is read from the dataset.

Based on whether the value is greater or smaller than a threshold (which is zero) the trigger is sent to one of two destinations.

The trigger plays one of these two speech samples every beat.

Here, dependent on the absolute deviation from 0, the gain is adjusted from a value of 30 to 127.

This transform the data to an absolute deviation from zero.

Figure 2. An example of the framework used to sonify Global Temperature Anomaly Data and CO2 Concentration Data.

dimensions, such as *Juggling Sounds* [28] also commonly make use of spatial playback methods to improve aesthetic results.

Spatial methods include simple panning, vector based-amplitude panning [29], altering source spectral content to control height perception [30] and controlling room size or distance through reverberation control [31].

4.6 Sampling

Many sonification methods do not seem to make much mention of sample usage, despite their popularity in other auditory display fields such as auditory icons, and their ubiquity in many forms of music. Sonification for aesthetic purposes obviously benefits from having a wide range of sound production methods at its disposal, and thus integration of sampling methods seems appropriate.

There are several sample triggering methods that are relevant. For stream monitoring purposes a sample may be triggered when a data stream passes a particular threshold, either in the negative or positive direction. This process is obviously useful for alerts etc., but may also be important for contextual information, such as presenting a sample associated with a particular time marker for instance.

Alternatively, by making use of the information contained in the bar and beat length controls used to control the timeline. With this information we can calculate the length of a bar and time-stretch a bar-long sample to fit within this bar. A rhythmical sample of some type can be used in this mode in exactly the same way as a sample.

The use of samples also invites the use of speech samples, seemingly prohibited by the definition of sonification mentioned in Section 1. However, an analogy with visual representations is useful in determining the place of speech in sonification. Many visual graphs succeed at representing large amounts of numerical data, where a tabular set of numbers would fail – but numbers and words are still included on the visual graph to provide context, or to annotate important or unusual parts of the graphic. Similarly, sonification can benefit from the use of speech annotation in auditory representations.

5. APPLICATION

To demonstrate the applicability of the toolkit we will discuss an example that sonifies global surface temperature anomaly data and CO2 concentration data.

The program in Figure 2 sonifies temperature deviation as either a 'hot' or 'cold' speech sample, using varying level. Simultaneously, the CO2 data is mapped to the pitch of a tone, using a quantisation to a number of pitches that present a chord. Each decade this quantisation set is changed, so that the chord progresses and harmonic momentum is established. This also serves an analytic function, helping to denote each decade as a different category. The example seems somewhat complex, but it is a one page visual program that describes the process of sonification in a relatively ordered fashion. It can be reordered arbitrarily, and allows extensive experimentation with aesthetic sonification techniques.

6. CONCLUSION & RESEARCH AGENDA

We have presented a toolkit for sonification that incorporates new methods geared towards aesthetics in auditory representation. A review of the literature describing such frameworks elucidated some of the tradeoffs that are inherent in these designs, but it also shows that few attempt to incorporate aesthetic purposes in their design.

Further research is underway to develop user interface objects that extend the framework with a touch-screen based user interface, featuring a spatial display mixer design, to allow interactive spatial mixing. Application to real-time data is planned, as are improvements in the data handling. Thorough experimentation and testing will assist in developing the framework and in developing new sonification techniques.

ACKNOWLEDGEMENTS

This research was supported by an Australian Research Council GrantDP0773107. The authors gratefully acknowledge the support of a Matsumae International Foundation Research Fellowship and Professor Koichi Hori, University of Tokyo (RCAST) A.I. Lab.

References

1. Walker, B.N. and G. Kramer, *Ecological psychoacoustics and auditory displays: Hearing, grouping and meaning making*, in *Ecological Psychoacoustics*, J.G. Neuhoff, Editor. 2004, Academic Press: New York.
2. Tufte, E.R., *Beautiful Evidence*. 2006, Cheshire, CT: Graphics Press.
3. Fry, B., *Computational Information Design*. 2004, MIT.
4. Tufte, E.R., *The Visual Display of Quantitative Information*. 1983, Cheshire, Conn: Graphics Press.
5. Moore, B.C.J., *An Introduction to the Psychology of Hearing*. 1997, London: Academic Press.
6. Zwicker, E. and H. Fastl, *Psychoacoustics: Facts and Models*. 1999, Berlin: Springer.
7. Walker, B.N., *Magnitude estimation of conceptual data dimensions for use in sonification*. Journal of Experimental Psychology: Applied, 2002. **8**(4): p. 211-221.
8. Neuhoff, J.G., G. Kramer, and J. Wayand, *Pitch and loudness interact in auditory displays: Can the data get lost in the map?* . Journal of Experimental Psychology, 2002. **8**(1): p. 17-25.
9. Joseph, A.J. and S.K. Lodha. *MUSART: Musical Audio Transfer Function Real-time Toolkit*. in *International Conference on Auditory Display*. 2002. Kyoto, Japan.
10. Lodha, S.K., et al. *MUSE: A Musical Data Sonification Toolkit*. in *International Conference on Auditory Display*. 1997. Palo Alto, California.
11. Wilson, C.M. and S.K. Lodha. *Listen: A Data Sonification Toolkit*. in *International Conference on Auditory Display*. 1996. Palo Alto, CA.
12. Walker, B.N. and J.T. Cothran. *Sonification Sandbox: A graphical toolkit for auditory graphs*. in *Proceedings of the 2003 International Conference on Auditory Display*. 2003. Boston, MA, USA.
13. Davison, B.K. and B.N. Walker. *Sonification Sandbox Reconstruction: Software Standard for Auditory Graphs*. in *Proceedings of the International Conference on Auditory Display*. 2007. Montreal, Canada.
14. Vickers, P. *Whither and wherefore the auditory graph? Abstractions and aesthetics in auditory and sonified graphs*. in *Proceedings of the 11th International Conference on Auditory Display*. 2005. Limerick, Ireland.
15. Campo, A.d., et al. *Sonification as an interdisciplinary working process*. in *International Conference on Auditory Display*. 2006. London, UK.
16. Campo, A.d., C. Frauenberger, and R. H öldrich. *Designing a Generalized Sonification Environment* in *International Conference on Auditory Display*. 2004. Sydney, Australia.
17. Worrall, D. *Overcoming Software Inertia in Data Sonification Research Using The SoniPy Framework* in *International Conference on Music Communication Science*. 2007. Sydney, NSW, Australia.
18. Worrall, D., et al. *SoniPy: The design of an extendable software framework for sonification research and auditory display*. in *International Conference on Auditory Display*. 2007. Montreal, Canada.
19. Yeo, W.S., J. Berger, and Z. Lee. *SonART: A framework for data sonification, visualization and networked multimedia applications*. in *International Computer Music Conference*. 2004. Miami, USA.
20. Yeo, W.S., J. Berger, and R.S. Wilson. *A Flexible Framework for Real-Time Sonification with SonART* in *International Conference on Auditory Display*. 2001. Espoo, Finland.
21. Ben-Tal, O., et al. *SonART: The Sonification Application Research Toolkit*. in *International Conference on Auditory Display*. 2002. Kyoto, Japan.
22. Pauletto, S. and A. Hunt. *A toolkit for interactive sonification*. in *International Conference on Auditory Display*. 2004. Sydney, Australia.
23. Barrass, S., *Auditory Information Design*. 1997, Australian National University.
24. Bevilacqua, F., R. Muller, and N. Schnell. *MnM: a Max/MSP mapping toolbox*. in *International Conference on New Interfaces for Musical Expression (NIME05)*. 2005. Vancouver, BC, Canada.
25. Schnell, N., et al. *FTM — Complex data structures for Max*. in *International Computer Music Conference (ICMC)*. 2005. Barcelona,.
26. Schnell, N. and D. Schwarz. *Gabor, Multi-Representation Real-Time Analysis/Synthesis*. in *COST-G6 Conference on Digital Audio Effects (DAFx)*. 2005. Madrid.
27. Song, H.J. and K. Beilharz, *Concurrent Auditory Stream Discrimination in Auditory Graphing*. Journal of Computers, 2007. **3**(1): p. 79-87.
28. Bovermann, T., et al. *Juggling Sounds*. in *International Workshop on Interactive Sonification*. 2007. York, UK.
29. Pulkki, V., *Virtual sound source positioning using vector base amplitude panning*. Journal of the Audio Engineering Society, 1997. **45**(6): p. 456-466.
30. Ferguson, S. and D. Cabrera, *Vertical localization of sound from multiway loudspeakers*, in *Journal of The Audio Engineering Society*. 2005.
31. Cabrera, D. *Control of perceived room size using simple binaural technology*. in *International Conference on Auditory Display*. 2007. Montreal, Canada.