A Toolkit for Interactive Sonification.

Conference Paper · January 2004

Source: DBLP

2 authors:

Sandra Pauletto
KTH Royal Institute of Technology

Andy Hunt
The University of York

A TOOLKIT FOR INTERACTIVE SONIFICATION

Sandra Pauletto & Andy Hunt

Media Engineering Group
Electronics Dept., University of York
Heslington, York, YO10 5DD, U.K.
{sp148,adh} @ ohm.york.ac.uk

ABSTRACT

This paper describes work-in-progress on an Interactive Sonification Toolkit which has been developed in order to aid the analysis of general data sets. The toolkit allows the designer to process and scale data sets, then rapidly change the sonification method. The human user can then interact with the data in a fluid manner, continually controlling the position within the set. The interface used by default is the computer mouse, but we also describe plans for multiparametric interfaces which will allow real-time control of many aspects of the data. Early results of interactive sonic analysis of two example domains are described, but extensive user tests are being planned.

1. INTRODUCTION

Data sonification reflects our everyday experience that the objects we interact with make sound. We have highly developed ear-brain systems which are used for decoding the world based on the sounds that it makes. For computer-based sonification to match these everyday experiences it is equally important to concentrate on the *interaction* methods as it is to consider the data-to-sound mapping. This focus on interaction in sound interfaces is dealt with more fully in the companion paper in this conference [1], and at the recent international workshop on Interactive Sonification [2].

Such real-time interaction is not limited to tasks where the data are being *gathered* in real-time. Even when the data has already been recorded, an element of user-interaction is needed to sift the data, tune the parameters, and focus onto the areas of interest. This is especially true of very large and complex, multi-dimensional data sets which need to be rapidly navigated to find areas of interest. This requires the sound to be rendered in real-time, and to react instantly to changes in the data. It is our belief that dynamic user-interaction with multi-parametric data streams can yield an effective method for controlling and interpreting those streams [3].

This paper describes a toolkit that is being developed at the University of York, UK, which allows the user to rapidly and interactively explore large data sets as sound, and to readily reconfigure the sonification algorithm being used.

2. BACKGROUND

In recent years a few sonification toolkits have been developed and presented to the research community. These toolkits are used to create an environment in which various sonification techniques can be explored, and different types of data experimented upon. There seem to be two basic approaches to creating such a toolkit. Either an application is programmed from scratch using a programming language such as C++, or new libraries that allow sonification are added to an existing environment that already has other useful characteristics. Examples of the first type of Toolkit are Listen [4], MUSE [5] and SonART [6]. An example of a sonification toolkit made up of new functions added to an existing environment has been implemented by Thomas Hermann in the visual environment system NEO [7]. The emphasis in the existing toolkits is on creating an application for experimenting with various sonification mappings. The researcher can map any data stream to various sound parameters and the sounds can be synthesized using many techniques. These toolkits also allow a certain level of interactivity. For example the user can vary the scaling or the mapping. In Hermann's work, data can also be navigated in real-time as it is mapped to sound [8].

With the Interactive Sonification Toolkit outlined in this paper, the emphasis is equally divided between providing many sonification techniques to experiment with, and providing several ways of interacting in real-time with the data as it is converted to sound. In particular, by real-time interaction with the sonification we mean either a) real-time navigation of pre-recorded data, or b) the mapping of data to sound *as* the data itself is generated.

3. INTERACTIVE SONIFICATION TOOLKIT

In this paper we present a prototype version of a toolkit which emphasizes the role of *interaction* with sonification algorithms. Therefore we needed to identify an environment suitable for the production of real-time sound synthesis and manipulation through various interfaces. The visual programming environment Pure Data [9] developed by Miller Puckette [10] has been chosen as the environment for creating the prototype of this toolkit due to the following reasons:

• it is cross-platform, working on PCs and Macintosh under Windows, OS and Linux operating systems;
• it allows real-time programming (no need for a compiling phase) which facilitates fast prototyping;
• it contains many built-in DSP (Digital Signal Processing) functions;
• it is freely available and open-source;
• it is extendible: new specific functions and libraries can be programmed in C and added to the environment.

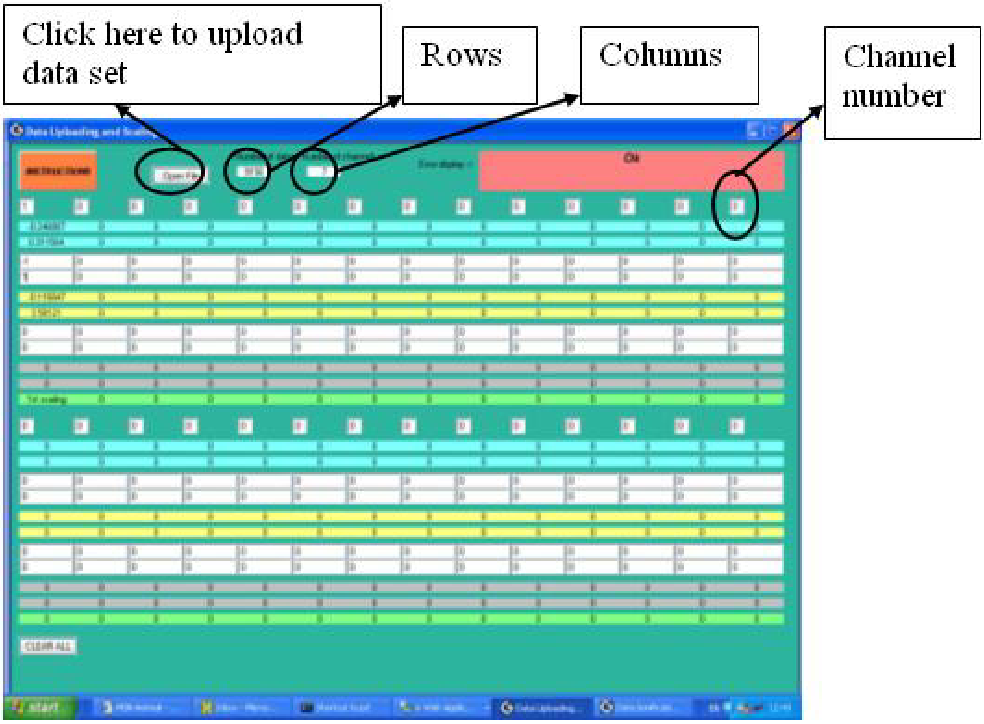

Figure 1: *Data Scaling page showing 28 channels*

The Toolkit makes use of existing PD libraries. In particular, it uses the basic PD objects that come with version 0.36.0, the Zexy [11] library 1.3 by Iohannes Zmoelnig, and the GriPD [12] library version 0.1.0 by Joseph A. Sarlo which is used to build the Graphical User Interface (GUI).

3.1. Data format and storage

A data set is represented by a matrix with columns and rows. In the case of time varying data sets, each column (also called a 'channel') can represent a different parameter, and the rows will be the measurements of these parameters at each instant in time. The number of rows in the matrix corresponds to the number of samples in each channel. The data set or matrix is written in a text file that can be edited using a simple text editor.
The first line of the file must contain the word 'matrix' followed by the numbers of rows and columns. A return separates the lines. Separators between values can be either spaces or tabs. The file has to be saved with the file extension '.mtx'.

3.2. Description of the Toolkit and its functions

The toolkit is divided into two modules that communicate with each other. This structure is reflected in the Graphical User Interface which has two main windows: the *data scaling* window and the *interactive sonification* window.

3.2.1. Data Scaling window

The function of this part of the program is to upload one or more data sets, and to scale them appropriately prior to sonification. The screen shows details of the data set(s) loaded (see screenshot in Figure 1).

Within this part of the program the user can scale each channel separately. The Toolkit allows two different methods of scaling:
• defining new minimum and maximum values;
• defining new transposition and stretching factors.

If the user employs the first scaling type, the program automatically calculates and displays the corresponding transposition and stretching factors, and vice-versa. Having the two types of scaling has proven to be very useful in the following situations:
1. When employing the second (stretching) scaling, the automatically generated minimum and maximum values can be used to determine whether the sound will be within the perceivable range.
2. When the user needs to display two channels using the same reference system (to allow direct comparison), then the second scaling is used. This ensures that both channels are transposed and stretched by the same amount. However if the user wants to ensure that the resultant values are still perceivable then this can be checked and altered using the first scaling.

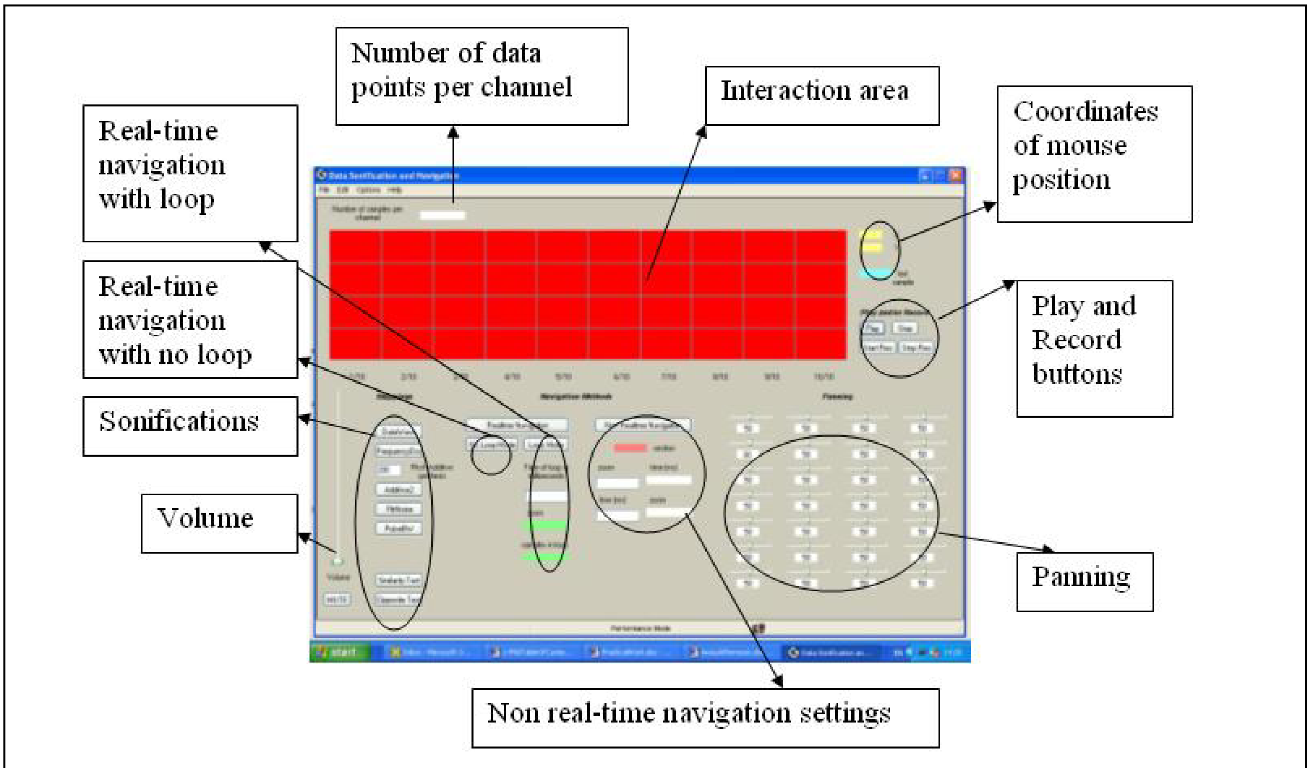

Figure 2: *Interactive Sonification page showing main functional areas*

3.2.2. Interactive Sonification window

After a data set has been appropriately uploaded and scaled, it appears in the Interactive Sonification window ready for conversion to sound, and navigation. From this screen (see Figure 2) the user can select the sonification method to be used and how to navigate through the sonified data. The user can navigate the data using the techniques described below.

3.3. Interactive Navigation

Within the toolkit the sound can be navigated in three different modes: two real-time and one non-real-time. The interaction area (see the rectangle within the screenshot in Figure 2) represents the sound in its totality (with time portrayed from left to right). It can be explored by dragging a mouse over it. The co-ordinates of the mouse are tracked by the program and used to generate sound. The navigation modes are as follows:

1) *Real-Time Navigation*: the user clicks the mouse and drags it within the interaction area. At each instant the co-ordinates of the mouse drive a pointer to the sound/data array. The sound is played in immediate response to the mouse movement, and therefore the user can freely move backwards and forwards through the sonification and hear the result. (The y axis of the interaction area does not have a function in this mode). Note that sound is only heard when the mouse is moving. This is rather like dragging a piece of recorded magnetic tape past a tape head (where sound is heard only when the tape is moving, and the rapidity of sonic events is determined by the speed).

2) Real-Time Navigation with Loop: this time the mouse x position controls the start-point of a sound loop. The user inputs the length of this loop prior to navigation. Even when the mouse is not moving, a sound can be heard, as the loop plays continuously. In this mode, the y axis has a zoom function: as the mouse pointer is moved down, the data is played through more slowly.

3) Non-Real-Time Navigation: in this case the user selects a zone of the data by clicking, dragging, and releasing the mouse. The x position of the initial click sets the start position, and the release sets the end. The user enters the required playback time and the program calculates the necessary zoom factor. To hear the sound the user presses the Play button. The sonic result of any navigation can be recorded as a .wav file and saved to disk for later analysis.

3.4. Types of sonification

A range of sonification methods has been implemented to date. These assorted methods aim to exploit what we know about human perception of sound. In particular, humans have the innate ability to analyse an auditory scene in such a way that some sounds are perceived as grouped together and others as separate. Such processes are referred to as the 'primitive processes of auditory grouping' [13].

The resulting sounds exhibit varied perceptual characteristics, sometimes eliciting sound grouping and at other times sound separation.
In particular, we identify five categories of sonification that can be made when implementing algorithms, so that

the resulting sounds are perceptually distinguishable. We have then tried to implement sonification types that cover all these possibilities. The identified categories are:

1) The data can be mapped onto *clearly recognisable perceptual parameters*, such as pitch, loudness and duration. The user can focus on the evolution of a particular parameter in order to gain an aural analogue of the data.

2) The data can be mapped onto sound variables so that the *overall timbre* of the sound represents the evolution of the data. Timbre is a complex sound phenomenon, different from parameters such as pitch. Many parameters and variables of a synthesis algorithm contribute to the composition of timbre.

3) When wishing to portray many channels of data, one can create a *mix of separately perceivable sounds*. For example assigning left-right panning values or different synthesised timbres to the different channels can improve their perception.

4) Many channels of data can contribute to the evolution in time of a *single sound* with a complex timbre.

5) In some cases *one* channel of data can be mapped onto *more than one* sound variable (possibly improving the perception of that data channel by utilising parameter redundancy).

Here follows the description of the implemented sonification types which allow us to experiment with all of the above mentioned characteristics. Eleven different types of sonifications have been explored to date. Eight are currently implemented in the Toolkit, while three exist as separate prototypes (see sonifications g, h and i).

a) **Audification**: the rescaled data points are read directly as sound samples. Many channels of data can be portrayed by mixing their individual audification. Panning can be applied to the different channels to aid channel separation.

b) **Oscillator frequency**: the scaled data points of a channel are fed into the frequency input of a wave-table oscillator, producing a sine wave with a pitch contour that varies with the data. The sound is continuous and many channels can be mixed. Different frequency ranges and panning values can be assigned to the various channels to improve separation.

c) **Additive Synthesis with constant pitch**: A sound is made by additive synthesis, with a constant fundamental frequency (chosen by the user). Each data channel drives the amplitude envelope of a particular partial. If only one channel of data is used, the result is the sound of a sine wave with changing loudness. If many channels are used the resulting sound can be perceived as a sound with a changing complex timbre.

d) **Additive Synthesis with variable pitch**: Again a sound is created using additive synthesis, but this time the data points of a specified channel control the fundamental frequency. The other channels drive the amplitude envelopes of the other partials, thus determining the moment-by-moment harmonic content. If only one channel of data is used, it can be mapped to both the fundamental frequency and the amplitude of the sine wave. If many channels are used a complex timbre is heard.

e) **Filtered Noise**: noise is synthesised and played through a band-pass filter. One data channel controls the centre frequency of the filter, while a second channel drives the filter's 'Q' factor (Centre Frequency / Bandwidth). If only one channel of data is used, it can be mapped to both the centre frequency of the filter and the 'Q' factor. A sound with changing timbre is heard, often perceived as a whistling effect.

f) **Pulse waveform**: one channel controls the frequency of a pulse waveform; while a second channel drives the pulse-width. Again, a single channel can be mapped to both variables.

g) **Data to note duration**: the data points of one channel control the durations of successive note events played on a MIDI synthesizer. The resulting sound consists of a stream of notes whose tempo represents the parameter under observation. This method will not work with multiple time-based streams because the synchronisation of data is destroyed by the mapping.

h) **Data to note pitch**: the data points of each channel drive the pitches of note events from a MIDI synthesizer. Each data channel is represented by a stream of notes. Different channels can be portrayed by mixing various streams together. Distinguishable timbres, different panning values and different registers (pitch ranges) can be assigned to the various channels to improve separation.

i) **Pitch and duration**: One channel is mapped onto note durations, the other onto note pitch. Either one data channel can be mapped to both pitch and duration, or two channels can be mapped onto pitch and duration respectively. In this case the result is a single sound stream portraying two channels.

The following two sonification methods were implemented to provide a quick way of comparing the evolution of two data sets.

j) **Similarity Test**: the data of two channels are normalized. One channel is then subtracted from the other (data point by data point) and the result (scaled to audible frequency range) is fed into the frequency input of an oscillator (see sonification type b). If the two channels' data are identical, then the frequency is zero and therefore no sound is heard.

k) **Opposition Test**: the data of two channels are normalized. The two channels are then added (point by point) and the result (scaled again to audible frequency range) fed into the frequency input of an oscillator. If the two channels' data are completely opposite to each other, then the frequency is zero and therefore no sound is heard.

4. APPLICATION AREAS

The toolkit is generally applicable to any data-set, but so far has been used specifically in two domains.

4.1. The generic system

The toolkit is generally applicable to situations where many simultaneous streams of data are gathered which then need to be analyzed to discover the underlying state of the system. In order to demonstrate the potential of the toolkit, we have selected two different application areas that have high-dimensional data sets which require exploration. These areas have been selected due to the remarkable similarity between the existing ways that their data is currently gathered and processed (see Figure 3).

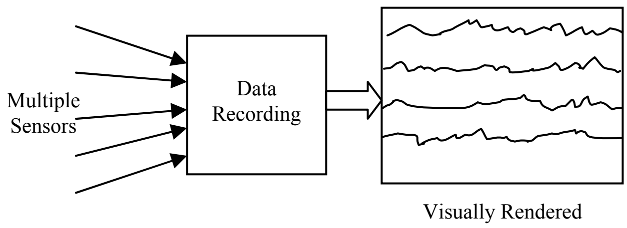

Figure 3: *Current visual analysis of gathered data*

Multiple sensors gather streams of data from different parts of the system. Together these data streams represent the state of the system. Currently in each domain the streams are displayed as numerical lists, or as individual graphs. Human analysts at present struggle to diagnose the relationships between the separate variables, and they have restricted methods at their disposal for moving through the data.

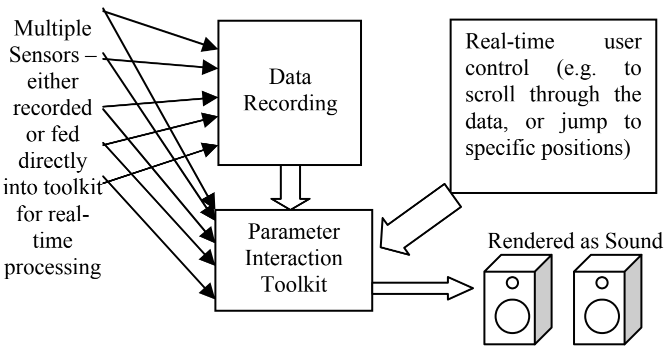

Figure 4: *System block diagram*

The Interactive Sonification Toolkit allows the user to access the recorded data streams and convert them into sound (see Figure 4). The human operator can then control the navigation through the data (for example scanning backwards and forwards) whilst listening to the resultant sonification.

We now explain how each of the chosen target application areas will utilize the toolkit.

4.2. Physiotherapy movement analysis

Therapists are being increasingly called to produce quantitative evidence of the progress of a particular condition (for example to assess the effectiveness of a particular form of treatment). In order to gather such numerical data, sensors can be attached to the surface of the body to monitor muscle potentials in various locations over time as a task is carried out (e.g. standing up). Vast amounts of data can be easily collected in this way, but the problem comes in trying to analyze and make sense of the many large streams of numbers. Therapists are convinced that the data streams hold the key to understanding the particular ailment, but they lack a tool for sifting, analyzing and manipulating the data.

The Interactive Sonification toolkit described here is the first stage of a project which plans to provide a diagnostic tool to aid therapists in assessing a complex set of movements. Such a tool will be used for helping to diagnose complex conditions, and monitoring the progress of individual patients.

Therapists will use a set of sensors and a data acquisition system to gather data for a particular movement. The analyst will use the Toolkit to parse that data (without the patient having to be in attendance) by listening to a sound-plot of individual variable streams, or the interaction of two or more streams (e.g. the difference in tension between two muscles). A further development will allow the system to be used in real-time by patients to enable them to monitor their own muscle movements (in sound) as they attempt to carry out specific movement tasks. The therapists would be able to give instructions such as "Try again, but this time try make the sound smoother", which is innately easier for most patients than "Try not to put too much tension in the largest muscle of your leg, compared with the smallest".

4.3. Helicopter flight analysis

Designers of aircraft regularly handle flight data and analyze it to aid the prototyping process. An on-line data acquisition system is used to gather flight data from the pilot controls and a large number of sensors around the aircraft under test. Large multi-parametric data sets are generated as a result, and these are currently examined off-line using visual inspection of graphs. The graphs are printed out and laid across an open floor to allow the whole data set to be seen at a reasonable resolution. Engineers walk around this paper array looking for anomalous values and discontinuities in the signal. We are working with engineers at Westland Helicopters to improve the methods for handling such data, with the immediate benefit of saved technician time, and thus a faster turnaround in flight data analysis. Flight data is converted into sound, and sifted through interactively by the human analyst. This helps to pinpoint problem areas due to, for example, local non-linear instabilities, undesirable dynamics and flight control system failures, and will give the analyst a better feel for the relationships between the data streams.

5. INITIAL OBSERVATIONS

A major aim of this project is to evaluate various interactive sonification methods. Two fundamental criteria will be used in such evaluation:
1) which mappings clearly portray the structures encoded in the data streams?
2) which mappings are best at portraying correlations between different data streams?
The following observations, based on the above criteria, were noted when using the prototype Toolkit with test data from the two example domains.

5.1. Audification

Data can be listened to very quickly using audification, typically (at CD sampling rate) 44100 data points per second are converted to sound. When using audification the human analyst can expect to hear:
• *a particular frequency* if there is periodicity in the data;
• *noise* if the numbers are randomly distributed;
• *increased loudness* for increased activity in the data;
• *nothing* if the data string is constant or moves smoothly and slowly;
• *a click* if there is a discontinuity in the data stream.

It is therefore possible to rapidly deduce some very basic, but fundamental, characteristics of the data stream just by listening to it as a continuous waveform.
When listening with this method to the physiotherapy data the muscles' firings are like bursts of noise with an amplitude and timbral envelope. It is clearly perceivable whether they are shorter or longer, louder or softer than expected. Our therapist collaborators are keen to investigate how the characteristics of the amplitude envelope and the overall timbre relate to specific muscle conditions.
The helicopter data set is much more complex in terms of the number of channels, and the huge numbers of samples. Sensors record the moment-by-moment values of different flight and user-control parameters (velocities, altitudes, movements of levers etc.). A typical half hour flight records the output of 400 sensors every 10 milliseconds – yielding a data file of 72 Megabytes. Yet using audification we can review each sensor's values over the entire flight in 4 seconds. Clearly noticeable are noise, discontinuities, the presence of a repetitive element, periodicity and amplitudes envelopes. The sounds produced have a complex structure, often a mixture of continuous smooth changes and noisy parts.
The *real-time navigation with loop* method was implemented to allow a sound to be heard even when the pointer is stationary. Clearly there will still be no sound if the data set is completely constant. Jumps and clicks can be located very easily with this method; the discontinuity is repeated regularly by the looping process and thus produces a harmonically complex signal, which is very easy to locate with the user control device.

5.2. Mapping data to frequency

Frequency is an important sound variable for mapping. The human ear is highly trained to track the pitch of sounds. People easily follow and remember frequency contours, and even communicate them vocally to other users. The pitch of the sound gives an easy-to-follow aural representation of the value of the data at each point. Smoothly changing data sets (which are barely audible under audification) are clearly heard as a gradual change of pitch. Jumps, noise and periodicities can be also heard as corresponding movements in frequency.

5.3. Mapping data to durations

Mapping to durations can be used to highlight certain patterns in the data. There are, though, limitations associated with this method. If the data strings are taken from a system which evolves naturally in time (such as our two example task domains!), then duration mapping destroys the analogical relationship between the data and time. Moreover, two channels mapped in this way cannot be mixed and compared simultaneously because they occupy different time scales.

5.4. Listening to simultaneous channels

Channels can be compared by playing them simultaneously. If the channels are mapped to different timbres of sound, the brain can follow the various streams separately. The use of stereo panning enables a greater audio separation of multiple channels [14].
This method has been used by the physiotherapists to verify a hypothesis that a particular set of supposedly symmetrical muscles were not in fact firing simultaneously. The sensors from the left side of a knee were panned left, and from the right side panned right. The resulting sound was rather like a complex stereo 'ping-pong' effect, demonstrating that when one muscle showed high activity the other did not, and vice versa. The physiotherapist working on this project found this fact very interesting because she had never seen a good visual display of this behavior, yet it was immediately obvious in the aural rendition.
Another approach to the sonification of two channels is to map the two data streams to variables that build the spectrum of the sound. The *Pulse* and the *Noise* sonification methods were implemented with this goal in mind. However, the parameters are mapped to two very different sound parameters (e.g. centre frequency and bandwidth of a filter) with different and non-comparable perceivable ranges. A careful study of what the perceivable changes in sound correspond to in the data needs to be carried out for this method to be a good display of the data.

When there are a large number of channels to be mapped onto many sound variables, it is a challenge to ensure that the data streams can be perceived separately. As mentioned above, it should be possible to synthesize a sound whose timbre changes in relation to the data. In this case, a certain timbre will become associated with a certain data configuration. We are working to find a way of representing the entire helicopter data (up to 28 parameters) set as a single complex timbre at each moment in time.

6. USER STUDIES & FURTHER WORK

A series of major user tests are currently under development. We plan to investigate how the nature of the *interface* and the *interaction style* affects the perception of the data in sound. A number of interface devices will be used in conjunction with the toolkit, to extend the range of interaction possible with the mouse.

6.1. User experiments

Experiments will be designed to compare various mappings using specially constructed data sets. These allow the researcher to know exactly which structures are present in the data, and thus to verify if they are perceivable. Experiments will also utilize real system data from our two target domains. The perception, by users, of any structure heard in the data will be verified using graphical and/or mathematical methods. Subjects will repeat the experiments on several different occasions so that observations can be made on their ability to learn and interpret each mapping. Various styles of data navigation will also be tested in order to explore the relationship between data mapping and data interaction.

6.2. Further sonification methods

Other ideas for future work include implementing new sonification methods using granular synthesis, subtractive synthesis and physical modeling. Vowel sounds can be produced by using subtractive synthesis (e.g. Klatt synthesizer [15]). By changing the values of formants a vowel-like sound changes into another sound with a very different tone 'color' (e.g. from the sound 'a' to the sound 'u'). The fact that humans are innately skilled at perceiving speech-like sounds makes this sonification method particularly worth investigating.

It could be very interesting to map data onto the parameters of physical models of sounding objects, (instead of directly onto sound parameters). Physical models offer both the promise of natural human interaction and many potential inputs. Many individual data streams could be mapped, for example, to the individual mass-points in a physical model.

Finally, a new parameter will be implemented in the Toolkit which will indicate to the user what the minimum perceivable step in the data is after a sonification method, and a particular scaling, has been chosen. This parameter will be a kind of 'resolution' indicator for the sonification.

At the conference we will present a new version of the Toolkit with added functionalities such as new sonifications of many channels and new interaction modes that use different interfaces. We also hope to present some initial experimental results.

7. CONCLUSIONS

This paper has described the first prototype Interactive Sonification Toolkit along with its initial informal results. Eleven sonification methods have been implemented. Early experiments have revealed structures in the data that were not at all obvious in the traditional visual-only analysis. Very large files can be listened to rapidly. We have begun to assess the relative advantages and disadvantages of the various sonification methods, and have noted how certain data structures are highlighted by different sonification mappings. The user interaction allows a more intimate understanding of the data, and we plan to analyze this in more detail with extensive user trials.

The next phase of the project will aim at improving the quality of the interaction by investigating multi-dimensional interfaces and the control of data-to-sound mapping. Experiments will be run to determine how successful each sonification method is at allowing certain structures in the data to be perceived.

8. ACKNOWLEDGEMENTS

The toolkit is being developed as part of the project 'Improved data mining through an interactive sonic approach'. The project was launched in April 2003 and is funded by the EPSRC (Engineering and Physical Sciences Research Council). The research team consists of academics at the Universities of York (User Interfacing and Digital Sound) and Teesside (Physiotherapy) led by Prof. Tracey Howe, and engineers at Westland Helicopters led by Prof. Paul Taylor.

9. REFERENCES

[1] Hunt, A. & Hermann, T. The importance of interaction in sonification, *Proc. ICAD* 2004.

[2] Hermann, T. & Hunt, A.D, The discipline of Interactive Sonification, *Proc. 1st Int. Workshop on Interactive Sonification,* Bielefeld, Germany, January 2004. http://www.interactive-sonification.org

[3] Hunt A. & Kirk R, Multiparametric control of real-time systems, (chapter) in *Trends in Gestural Control of Music*, eds. Marc Battier, Joseph Rovan & M. Wanderley, FSU - Florida, (April 2000).

[4] Wilson C. M., Lodha S. K., Listen: a data sonification toolkit, *Proc. ICAD* 1996

[5] Lodha S. K., Beahan J., Heppe T., Joseph A., Zane-Ulman B., MUSE: a musical data sonification toolkit, *Proc. ICAD* 1997 pp. 1119-1134, 1997

[6] Ben-Tal O., Berger J., Cook B., Daniels M., Scavone G., Cook P., SONART: the sonification application research toolbox, *Proc. ICAD* 2002

[7] http://www.techfak.uni-bielefeld.de/ags/ni/projects/simulation_and_visual/neo/

[8] Thomas Hermann, Jan Krause, and Helge Ritter, "Real-time control of sonification models with an audio-haptic interface," *Proc. ICAD* 2002, R. Nakatsu and H. Kawahara, Eds., pp. 82–86.

[9] Pure Data: http://www.pure-data.org

[10] Miller Puckette: http://www.crca.ucsd.edu/~msp/

[11] Zexy externals: http://www.pure-data.org/downloads/

[12] Gripd: http://crca.ucsd.edu/~jsarlo/gripd/

[13] Bregman, A. S., *Auditory Scene Analysis*, MIT Press, 1994

[14] Brown, L., Brewster, S., Ramloll, R., Yu, W. & Reidel, B., Browsing Modes for Exploring Sonified Line Graphs, Proc BCS HCI 2002.

[15] Klatt, D. H., 1980, Software for a cascade/parallel formant synthesizer, *Journal of Acoustical Society of America*, 67:3, pp. 971-995