MUSART: MUSICAL AUDIO TRANSFER FUNCTION REAL-TIME TOOLKIT

Abigail J. Joseph *Suresh K. Lodha*

Department of Computer Science
University of California
Santa Cruz, CA 95064
{ajjoseph,lodha}@soe.ucsc.edu

ABSTRACT

This work describes the design and implementation of a sonification toolkit. MUSART (MUSical Audio transfer function Real-time Toolkit) is a sonification toolkit which produces musical sound maps that are played in real-time. Register, pitch, timbre, chords, duration, silence, loudness, beats, and panning are the musical concepts used to create melodic sound maps. Univariate and multivariate data sets are sonified using various sound parameter combinations and music tracks. Users have the flexibility to create personalized auditory displays by mapping each data dimension of a data set to one or more sound parameters.

MUSART is designed to be flexible so that it can be used with many applications. In this work, we use musical auditory maps to explore seismic volumes used for detecting areas for drilling oil.

1. INTRODUCTION

Audio mapping techniques provide the foundation for sonification toolkits that help people create sound maps. Approaches to creating auditory mappings include synthesized, natural/event-based, and musical techniques. This research focuses on using musical techniques to generate engaging, non-fatiguing, and familiar sounds and allow for meaningful exploration of information. MUSART uses a systematic approach to building musical sound maps. The tool generates basic sound maps that can also be customized for complex uses by using music theory as a foundation for sound maps and using a graphical legend of sound parameters as a data mapping interaction technique. Users can manipulate ten musical elements: register, pitch, duration, silence, loudness, thickness, timbre, balance, beats, and consonance to create a unique auditory display of univariate or multivariate data.

MUSART was developed using audio transfer functions and music theory constructs. We incorporate audio transfer functions into our sonification tool to provide an overall view of a system in which there is a virtually limitless number of sonification parameters used to describe data and which are defined by the user, and to create a complementary graphical representation of the relationship between sound parameters, which also allows the fine-tuning of parameters.

Music is the science or art of incorporating intelligible combinations of tones into a composition having structure and continuity. We use music theory concepts in our sonification tool to: generate sounds that are musical and familiar; design sound maps ranging from simple (e.g., varying pitches) to complex (e.g., more than one music track) that can accommodate for various users and applications; and have sound parameters with a discrete number of levels and that require no knowledge of sound synthesis to understand how they function together. The combination of transfer function and musical concepts creates an environment that allows any person to generate the most simple or complex auditory displays regardless of music, scientific, or sound synthesis background.

2. RELATED WORK

For many applications, visual displays do not satisfactorily present data in a format that promotes precise information extrapolation by a user. Auditory displays, however, can allow for rapid detection of complicated information, orient a user to key data, and promote ease of learning an application, to a name a few uses [1]. The main component of an auditory display is auditory mapping.

We classify auditory maps by the types of sounds used to generate the sonification scheme: synthesized [2], natural/event-based sounds [3], and musical [4]. In comparison to synthesized and natural/event-based mappings, *musical* mappings are generated with the belief that sounds which have some relationship to one another are more interesting, pleasing, and meaningful, as in music. Music theory has proved to be useful in developing auditory maps in many sonification applications [5, 6, 7, 8, 9, 10]. However, limited research has been focused on the development of a general purpose sonification toolkit which can exploit musical constructs to provide a flexible and general environment for generating musical sound maps.

Several efforts have utilized synthesized sounds and arbitrarily combined music constructs in sonification toolkits [11, 12, 13]. The use of complete musical theory and music constructs has been limited, although musical sounds do appear in many application specific auditory mappings. Barrass and Robertson developed a perceptual sound space as a method for standardizing the capabilities of sound display [14]. The authors developed a sound model that was limited to three sound parameters: timbre, brightness, and pitch. MUSE, another sonification toolkit having three more sound parameters than did the Barrass and Robertson perceptual sound space, strove to create meaningful mappings using melodic constructs and musical concepts [15]. MUSE is a musical sonification environment which generates musical sounds and allows mapping of data to sound parameters, such as timbre, rhythm, volume, pitch (melody), tempo, and harmony. The toolkit also provides the user with the ability to map sound to both univariate and multivariate data sets. However, the complexity of the musical concepts combined together can make it difficult to extract meaningful information through multivariate sound mapping.

MUSART improves upon other sonification toolkits by: introducing a large pool of sound parameters that combine in a systematic manner; providing a graphical legend of the relationships between sound parameters which extends data mapping interaction beyond buttons, text input, and sliders; and providing a way in which to overload sound parameters to more than one data variable.

3. MUSART

3.1. Overview

MUSART is a general purpose sonification tool that uses audio transfer functions to map data to sound [16]. The sonification tool allows a user to explore either univariate or multidimensional datasets through sound by generating audio maps that incorporate several music concepts. Music parameters (register, pitch, timbre, thickness, duration, silence, beats, balance, and consonance) provide the basis for these sound maps and combine to form musical notes.

3.2. Audio Transfer Functions (ATFs)

A transfer function defines the relationship between the inputs to a system and its outputs and provides a continuous mapping of data. Initially the transfer function graph of an ATF is empty. Data values mapped to an empty function produce identical musical notes. All notes (sounds) are defined by the default sound parameters values (see Section 3.4 for details on these values), unless otherwise defined by the user, who can manipulate the function curve in the graph. To reduce clutter, function curves are introduced into the ATF graph only as needed by the user. In the most complex scenario, an ATF graph can contain up to as many function curves as there are data parameters (10) times the dimensions of data. For example, a multivariate data set of 4 dimensions can have a transfer function graph with as many as 40 function curves. Both univariate and multivariate data can be mapped to sound using this scheme. Depending on the type of data set, the data can be sonified in one of the following ways:

1. For a one dimensional data input stream, X, and an audio transfer function defined by the audio parameter function curves, $A, B, C, ...J$, the following shows how a sound for a specific data value is calculated:

$$Data(x_i) \Longrightarrow \mathbf{ATF} \Longrightarrow Sound(A(x_i), B(x_i), C(x_i), ..., J(x_i)).$$

2. For a multidimensional data set, with data input streams, $X_1, X_2, X_3, ..X_n$ representing n dimensions of the data, and an audio transfer function with parameter curve functions, $f_1, f_2, f_3, ..f_n$ associated with each respective data dimension, the sound for a multivariate data value is calculated as follows:

$$Data(x_{1i}, x_{2i}, x_{3i}, ..., x_{ni})) \Longrightarrow \mathbf{ATF} \Longrightarrow Sound(f_1(x_{1i}), f_2(x_{2i}), f_3(x_{3i}), ..., f_n(x_{ni})).$$

3.3. Implementation

MUSART has been created using C++, Csound, Fltk, and OpenGL. It has three main parts: 1) a graphics window for building a transfer function, 2) a user interface for adding function curves, and 3) an interface for playing the sound generated from the auditory map. Figure 1 shows the graphical user interface for MUSART.

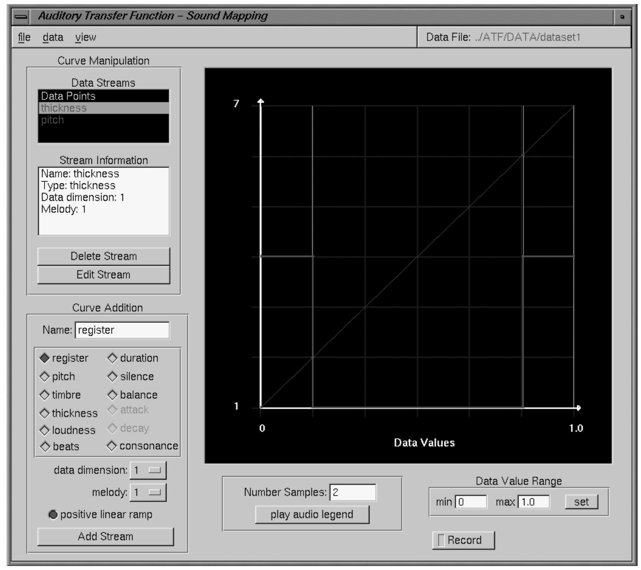


Figure 1: MUSART user interface. Transfer function editing widow is on the right. Interface for adding function curves is on the left.

3.4. Music Elements

Ten musical elements are implemented as the foundation for generating sounds in MUSART, including: **register** (an octave or a range of pitches); **pitch** (the frequency of a sound); **timbre** (the general prevailing quality or characteristic of a sound); **thickness** (the addition of adjacent frequencies to an existing frequency to create a combination of tones that blend together); **duration** (the length of time a sound is heard); **silence** (the length of time no sound is heard); **loudness** (describes the strength of the ear's perception of sound); **balance** (the location of sound between the right and left ear); **beats** (the rapid repetition of a musical tone which causes the notes to sound with rapid pulses); **consonance** (a combination of tones that is pleasing to the ear and produces a feeling of stability).

Each element is broken into levels which are used to create functions curves in the ATF graph. Each element is derived from basic tonal Western music theory. and can be represented by music notation. Table 1 provides a summary of each element's unit (e.g., frequency (Hz)), levels, and default value. These musical elements can be implemented alone or in conjunction with one another to form simple or complex sound mappings.

3.5. Creating sound maps

A user generates a sound map by specifying a transfer function graph. A transfer function graph is built by the addition of transfer function curves that relate to particular data attributes. Sound parameters are chosen by the user based on the characteristics of a particular dataset and the user's musical construct preferences. A transfer function graph can be as simple as a linear ramp of one curve or as complex as numerous curves with complicated shapes. The complexity of function curve shapes is determined by the user. An audio map is sonified by associating a certain number of sequential data samples or a data set with the ATF.

Element	Unit	# Levels	Values	Default Values	Utilized Theory
register	-	5	octave2 - octave6	octave 3	octaves
pitch	Hertz (Hz)	12	note names [A, A#(Bb), B, C, C#(Db), D, D#(Eb), E, F, F#(Gb), G, G#(Ab)] frequencies	C (261.62Hz for octave3)	equal tempered musical scale
timbre	-	9	strings, brass, flute, clarinet oboe, piano, trumpet, drum, bell	strings	instruments
thickness	-	7	1 note - 7 notes	1	chords
*duration	seconds (sec)	12	whole, half, quarter, 8th, 16th, 32nd	quarter note (1 sec)	note lengths
*silence	seconds (sec)	13	whole, half, quarter, 8th, 16th, 32nd, none	no rest (0 sec)	rest lengths
loudness	decibels (dB)	7	30, 40, 50, 60, 70, 80, 90	60dB	sound intensity levels
balance	-	5	left, center, right	center	panning filter
beats	-	10	1 - 10	1	rapid repetition of notes
consonance	-	13	consonant - dissonant	consonant (unison)	2 notes in combination

Table 1: Summary of musical elements used in MUSART for building function curves. *For each note value, there exists a dotted note value (1.5 x) that proceeds it.

3.6. Generating sound

Data can be mapped to an audio map and sonified. In addition, a sound legend can be sonified to preview the effects the sound mapping would have on data before an entire data set is sonified. Each sound parameter has a default value, which can be changed by the user. For the sound parameters not defined by a function curve in the transfer function graph, the default parameters define those characteristics of a note.

3.6.1. Sonification of the Audio Legend

The sound map created by the transfer function can be listened to by selecting the number of samples (from 1 to n) to be played. These samples constitute an *audio legend*. When the audio legend is played, each sample being sonified is displayed as a vertical line in the transfer function graph. A user can also select specific data values to sonify one at a time. Another way to sonify data values in the audio legend is to choose continuous sound play and to manually move a slider through the data points.

3.6.2. Sonification of Data

The program can sonify two types of data, univariate and multivariate. Data values are played sequentially while they are read from a data file. When a univariate data value is sonified, this value is displayed in the transfer function graph as a vertical line. Data tuples are played sequentially as they are read in from the data file. When a tuple is sonified each data dimension value is visually displayed simultaneously as a vertical line in the color of the corresponding function curve.

3.6.3. Medit

An additional aspect of the program called **Medit** (music editor) can be used to explore the features of MUSART. Medit plays notes continually and changes the characteristics of the note as the user manipulates the sound parameters with a slider interface. This provides users with a way in which to decide which sound parameters to assign to which data values or attributes.

4. EXAMPLES

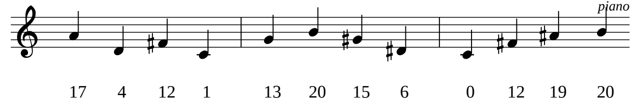
Sound maps can be utilized for practical data representation. Sound maps are flexible and can present much data simultaneously because of the number of different parameters available. Because MUSART has so many parameters, it also allows for user-defined preferences, making it versatile for both many applications and for diverse users. The following sections explore scenarios that use simple data sets of 12 variables ranging in value from 0 to 20.

4.1. Univariate Data Mapping

In this example the dataset consists of 12 data with the values: 17, 4, 12, 1, 13, 20, 15, 6, 0, 12, 19, and 20. Using MUSART there are numerous ways in which a sound map can be created for the univariate data stream. In all of the following scenarios one track (one instrument used to describe a sequence of notes) is used to represent the data stream.

4.1.1. Scenario 1 - Simple Mapping

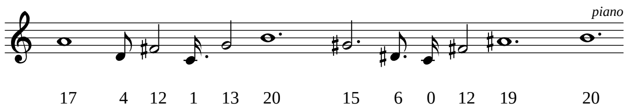
In the first scenario a sound map is created using one varying sound attribute (pitch). Data is mapped as a linear pitch mapping using quarter notes and piano timbre, where lower data values have a lower pitch and higher values have higher pitches. The result is as follows:



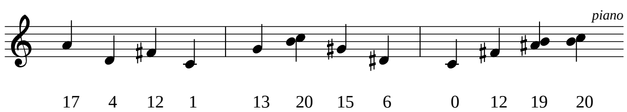
When listening to this data, a user would be able to discern differences among data as well as patterns based on the levels of pitch.

4.1.2. Scenario 2 - Emphasizing Values

More complicated sound maps can be created using multiple sound parameters to represent the univariate data stream. Introducing a second sound parameter can emphasize the characteristics of the data stream, as well as indicate areas of interest. For example, duration can be combined with pitch to emphasize the information provided by a simple pitch mapping. In this example, higher data values are mapped to long durations and high pitches, and lower data values are mapped to short durations and low pitches. The result of mapping the 12 data values is:



Certain data (e.g., higher data values) can be emphasized by a second or even third attribute, such as consonance. This trace uses a linear mapping of pitch, where higher values are mapped to high pitch and lower values are mapped to low pitch in combination with consonance, where of the highest 10% of the values are mapped to the highest level of consonance (dissonance) and the remainder of the values are mapped to the lowest level of consonance (which is the original note). The result is as follows:



4.2. Multivariate Data Mapping

The following examples explore some of the numerous ways that MUSART can represent multidimensional data with sound. The data for this example consists of a multivariate dataset of 12 variables of 3-tuples. Each dimension might correspond to data attributes such as temperature, velocity, and mass, and to sound map attributes such as pitch, duration, or silence. The data set is shown in Table 2.

4.2.1. Scenario 1 - Simple Mappings

In the first scenario we explore the multidimensional data using three sound parameters and one track to describe the sound aurally. First we map Dimension 1 to pitch, Dimension 2 to duration, and Dimension 3 to silence using linear ramp curves. Low data values are mapped to low pitches for Dimension 1, short durations for Dimension 2, and short rest lengths for Dimension 3. High data values are mapped to high pitches for Dimension 1, long durations for Dimension 2, and long rest lengths for Dimension 3. The result is as follows:

In this case Dimension 1 and Dimension 2 are played simultaneously as a single note of a certain pitch and duration, and Dimension 3 is shown by the silence (rest) between notes.

A sound map does not have to utilize all the provided dimensions of data; similarly, one dimension can be emphasized by assigning it to more than one sound parameter. In the next case the sound map includes only dimensions 1 and 3. Using linear ramps Dimension 1 is mapped to pitch and duration, and Dimension 3 is mapped to balance. Low data values are mapped to low pitches and short durations, and high data values are mapped to high pitches and long durations for Dimension 1. For Dimension 3, the low range data values are mapped to left balance, mid-range data values to center balance, and high values to right balance.

4.2.2. Scenario 2 - Multiple Tracks with Same Timbre

In this scenario each data dimension is represented by a separate track. Each data dimension is played using a linear pitch scale, where a higher pitches equate to high values and lower pitches equate to low values. The piano timbre is used for all three dimensions. When played in combination the tracks sound like a series of chords. When the values differ among dimensions a dissonant chord is struck; when the dimensions equal one another, only one note is heard, although in reality two or three identical notes are being played simultaneously. The results are as follows (3 identical notes played simultaneously are highlighted with a *):

4.2.3. Scenario 3 - Multiple Tracks with Different Timbres and Balance

In this scenario the multivariate data is mapped using two tracks. Dimensions 1 and 2 are mapped to Track A using a piano timbre, a linear ramp pitch curve, and a thickness threshold curve. The audio is played in the left stereo channel. Low data values are mapped to low pitch, and high data values to high pitch for Dimension 1. For Dimension 2 low and high range data values are mapped to thickness level 1 and mid-range data values are mapped to thickness level 3. Dimension 3 is mapped to a separate track, B, played in the right stereo channel using drum timbre and a linear ramp beats curve. Low data values are mapped to low beats levels and high data values are mapped to high beats levels. The results are as follows:

Variable	1	2	3	4	5	6	7	8	9	10	11	12
Dimension 1	17	4	12	1	13	11	15	6	2	12	19	20
Dimension 2	12	6	12	8	15	15	0	9	2	1	20	0
Dimension 3	6	3	12	20	16	11	11	8	2	10	8	20

Table 2: Multivariate data set of 3 dimensions.

5. APPLICATION: SEISMIC DATA

We used MUSART to create audio representations of 3D seismic data. In this research we focus on the visualization of sedimentary basins, where the imaging of rocks and layers is used for the discovery of oil and gas. We will explore how sonification can be useful to interpret and analyze 2D visualizations of slices of sedimentary layers generated from 3D seismic surveys.

5.1. Visualization

Researchers distinguish rocks and layers of soil by the differences in the velocity waves generated by seismic reflective exploration equipment. Travel-time, amplitude, and frequency of reflected and refracted waves are the primary data sources for seismic exploration. The data collected from reflection surveys are used to generate visualizations of the rock and layers of the earth's interior.

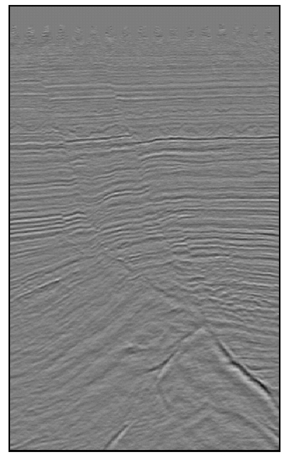
Currently, graphical computer applications are used to create visual images of the data collected from seismic surveys. Visualizations are interpreted and analyzed by human seismic data interpretors. In a 3D seismic data set, anywhere from 50-70 attributes need to be analyzed to make assessments of a site. However, no more than two attributes can be visualized at one time to convey useful, non-muddled information. Auditory display, on the other hand, allows for the simultaneous analysis of multiple attributes.

5.2. Sonification

The representation of seismic data in exploration seismology is constantly evolving. In the past decade the use of auditory displays has been explored as a way to enhance visual representations of geological data. Saue discusses a framework for incorporating and interacting with sound in visual environments [17]. The focus of his research is on large spatial data sets (i.e., seismic data and ultrasound images), but provides no specific examples for how the framework is applied to seismic data. Barrass and Sehner use a virtual Geiger-counter sound metaphor to represent multivariate well log data in conjunction with a virtual visual display [18]. The Virtual Geiger metaphor incorporates 10 granular synthesis parameters to distinguish between data on local, intermediate, and global levels, where attributes differ in timbre. The authors demonstrate the usefulness of the Virtual Geiger metaphor as it applies to well-log data, however this is just one facet of geological data that can be explored with sound. We focus on developing auditory displays to describe the multidimensional data used in seismic data analysis.

5.3. Results

Currently we use MUSART to incorporate sound into 3D seismic visualizations to aid in data analysis. A typical visualization is a 2D slice of a 3D seismic data cube (See Figure 2). Each pixel in



(a)

(b)

Figure 2: YZ slices of seismic volume attributes: (a) amplitude – low amplitudes are white, mid amplitudes are gray, high amplitudes are black; (b) amplitude difference – low difference are white, high differences are black.

the image represents one data attribute (i.e., amplitude or amplitude difference). We use sound to represent other data attributes that can aid in the interpretation of the 2D slices of seismic data. MUSART provides continuous and discrete exploration of seismic data with sound for detecting hidden patterns and anomalies. Auditory display is used to represent data points, traces (two or more data points sonified in combination), and ratio of local maxima.

5.3.1. Fault Exploration

A *fault* is a rock fracture along which movement or displacement in the plane of the fracture has taken place. In a visual display faults are distinguished by breaks or discontinuities in rock layers. Faults are indicated in areas where the amplitude difference is large. A sound map using drum timbre, 8th notes (0.75 second durations), and a beats function curve is used to sonify the potential fault. The audio of the path highlighting the true fault is rapid with high beat levels constantly played one after the other. In contrast, the sonified path not highlighting an actual fault is slow in tempo with low beat levels. Faults are thus represented by constant, rapid beating.

5.3.2. Layer Consistency

Rock layer classification is another aspect of seismic data analysis. Visualizations of seismic data show rock layer stratification in the earth's surface. Sonification can be used to re-enforce observations of rock layer consistency in an area. In this example traces of 4 sonified amplitudes show the consistency or inconsistency of rock layers. The sound map used for this example consists of 4 tracks, each using piano timbre, 16th notes (0.250 second durations), and the pitch and register function curves. The sound mapping is played on a separate track for each amplitude value in the trace. The consistent layer generates chords of 4 pitches that are mostly consonant. Inconsistent rock layers, on the other hand, will generate sounds that are dissonant, thus indicating that the values within a trace are dissimilar.

5.3.3. Ratio of Local Maxima

The ratio of local maxima provides information about the Gaussian distribution of amplitudes in a seismic data set, particularly about the values above and below a path. The sound map consists of 3 audio tracks, each using a linear pitch function curve with 16th notes (0.25 second durations). The first track uses the flute timbre and is accessed when the maximum above is larger than the maximum below. The second track uses the piano timbre when the maximum above equals the maximum below. The third track uses the drum timbre and signifies when the maximum above is smaller than the maximum below. Finally, no sound is played if a ratio is not computed for a voxel.

6. CONCLUSIONS AND FUTURE WORK

We created a flexible, extensible sonification toolkit that can generate both simple and complex sound maps. MUSART combines music concepts with transfer function theory to provide a variety of sonification parameters, provides a graphical representation of sound mappings, and produces interesting and musical sound maps for univariate and multivariate data. We have shown examples of the numerous possibilities for mapping data with sound through audio transfer functions.

We are currently applying auditory maps to a few applications for information analysis. We are researching the use of auditory display to better analyze three-dimensional seismic data and to detect areas for drilling oil, as well as looking at the role sound can play in exploring flow in vector field data. MUSART has been purposely designed to be flexible, extensible, and applicable to any area a user may wish to explore.

7. ACKNOWLEDGEMENTS

The authors are indebted to the members of the Chevron Geoscience Visualization group for providing 3D seismic datasets, as well as their feedback and discussion on sonification of seismic data. We also thank Professor David Jones at UCSC for his suggestions and feedback on music theory.

8. REFERENCES

[1] G. Kramer, “An introduction to auditory display,” in *Auditory Display, Sonification, Audification, and Auditory Interfaces*, G. Kramer, Ed., pp. 1–77. Addison-Wesley, 1994.

[2] E.S. Yeung, “Pattern recognition by audio representation of multivariate analytical data,” *Analytical Chemistry*, vol. 52, pp. 1120–23, 1980.

[3] W.W. Gaver, “Auditory icons: Using sound in computer interfaces,” *Human-Computer Interaction*, vol. 2, pp. 167–177, 1986.

[4] M.M. Blattner, D.A. Sumikawa, and R.M. Greenberg, “Earcons and icons: their structure and common design principles,” *Human-Computer Interaction*, vol. 4, no. 1, pp. 11–44, 1989.

[5] G. Mayer-Kress, R. Bargar, and I. Choi, “Musical sturctures in data from chaotic attractors,” in *Auditory Display, Sonification, Audification, and Auditory Interfaces*, G. Kramer, Ed., pp. 341–368. Addison-Wesley, 1994.

[6] J.L. Alty, “Can we use music in computer-human communication?,” in *People and Computers X*, D. Diaper and R. Winder, Eds. Cambridge University Press, 1995.

[7] Paul Vickers and James L. Alty, “Caitlin: A musical program auralisation tool to assist novice programmers with debugging,” in *International Conference on Auditory Display*, 1996.

[8] U. Axen and I. Choi, “Investigating geometric data with sound,” in *International Conference on Auditory Display*, 1996, pp. 25–28.

[9] J.L. Alty and D.I. Rigas, “Communicating graphical information to blind users using music: the role of context,” in *Human Factors in Computing Systems*, April 1998, pp. 574–581.

[10] S.K. Lodha, D. Whitmore, M. Hansen, and E. Charp, “Evaluation of a musical-visual system: Does music make any difference?,” in *International Conference on Auditory Display*, April 2000, pp. 167–172.

[11] P. Astheimer, “Sonification tools to supplement dataflow visualization,” in *Scientific Visualization: Advanced Software Techniques*, P. Palamidese, Ed., chapter 1.2, pp. 15–36. Ellis Horwood, 1993.

[12] T.M. Madhyastha and D.A. Reed, “Data sonification: Do you see what I hear?,” *IEEE Software*, vol. 12, no. 2, pp. 45–56, March 1995.

[13] C.M. Wilson and S.K. Lodha, “LISTEN: A data sonification toolkit,” in *Proceedings of the International Conference on Auditory Display ’96*, pp. 35–40. ICAD, November 1996.

[14] S. Barrass and P.K. Robertson, “Data exploration with sound using a perceptually linearized sound space,” in *Proceedings of the SPIE - The International Society for Optical Engineering*, 1995, vol. 2410, pp. 313–322.

[15] S.K. Lodha, J. Beahan, T. Heppe, A.J. Joseph, and B. Zane-Ulman, “Muse: A musical data sonification toolkit,” in *Proceedings of the International Conference on Auditory Display ’97*, pp. 61–64. ICAD, November 1997.

[16] A.J. Joseph, *MUSART: Musical Audio Transfer Function Real-time Toolkit; A Musical Approach to Sonification*, Ph.D. thesis, University of California - Santa Cruz, 2002.

[17] S. Saue, “A model for interaction in exploratory sonification displays,” in *International Conference on Auditory Display*, 2000.

[18] S. Barrass and B. Zehner, “Responsive sonification of well-logs,” in *International Conference on Auditory Display*, 2000.