MUSE: A Musical Data Sonification Toolkit

Suresh K. Lodha, John Beahan,[*] Travis Heppe, Abigail Joseph, and Brett Zane-Ulman
Department of Computer Science
University of California, Santa Cruz, CA 95064
lodha@cse.ucsc.edu
Tel: (408)-459-3773
Fax: (408)-459-4829

ABSTRACT
Data sonification is the representation of data using sound. Last year we presented a flexible, interactive and portable data sonification toolkit called LISTEN, that allows mapping of data to several sound parameters such as pitch, volume, timbre and duration [20]. One of the potential drawbacks of LISTEN is that since the sounds generated are non-musical, they can be fatiguing when exploring large data sets over extended periods of time. A primary goal in the design of MUSE -- a MUsical Sonification Environment -- is to map scientific data to musical sounds. The challenge is to ensure that the data meanings are preserved and brought out by these mappings. MUSE provides flexible data mappings to musical sounds using parameters such as pitch (melody), rhythm, tempo, volume, timbre and harmony. MUSE is written in C++ for the SGI platform and works with the freely available sound specification software CSound developed at MIT. We have applied MUSE to map uncertainty in some scientific data sets to musical sounds.

Keywords: CSound, harmony, melody, music, rhythm, sonification, timbre.

INTRODUCTION
The usefulness of integrating sound into visualization systems is well-recognized [13]. Several examples of applying sonification to scientific data analysis in diverse disciplines such as chemical analysis, economic analysis, analysis of algorithms, seismology, and computational fluid dynamics have been reported [8]. However, most sonification systems currently use pre-recorded patterns or simple synthesized sounds, which can become irritating or fatiguing due to their unchanging nature. This makes it difficult to explore large data sets or small data sets over extended periods of time using sonification.

In this work, we present MUSE -- MUsical Sonification Environment -- for sonifying data. MUSE generates musical and engaging sounds allowing interactive and flexible mapping of data to six different sound parameters -- timbre, rhythm, volume, pitch (melody), tempo and harmony.

BACKGROUND
In this work, we use sound in conjunction with visualization. An excellent discussion of the benefits of this approach can be found in Kramer [14].

Although several sonification systems have been proposed [2, 3, 11, 17, 16, 15] that can work in conjunction with visualization, most of these systems use a finite collection of pre-recorded sounds or very simple synthesized sounds. This results in two major drawbacks. First, since the collection of different sounds that can be generated is rather small, the flexibility in mapping data to different types of sounds is limited. Second, the repetitive and artificial nature of these sounds can become irritating and distracting, causing fatigue.

Some of the early attempts to generate a wider collection of sounds by using sound synthesis algorithms are reflected in works by Gaver [10] and Scaletti [18]. The NCSA audio development group at the University of Illinois, Urbana Champaign has proposed sound specification for immersive virtual environments, such as the CAVE, using computational models [5]. However, these sounds are still non-musical. More recently, the same group has been involved in the development of musical sounds for specific applications such as the exploration of biological data sets in the CAVE [4] and alpha-shapes [1]. Our approach differs from theirs in at least two main aspects. First, our objective is to develop a general purpose musical data sonification toolkit that can be applied to a wide variety of applications. In particular, the mapping of

[*] Department of Music

data to timbre that has been used by Brady et al. [4] is one of the several functionalities that MUSE can deliver. Second, we use the very powerful and widely available CSound as the sound specification language.

CSound

CSound is a computer program for the specification of sound [19] developed at MIT in late 1980s. For this project, we chose to work with CSound because it is (i) flexible, (ii) free and easily available through internet, (iii) available on many platforms, and (iv) well supported through the specification of many well-known CSound instruments [9]. CSound itself is written in C. Moreover, CSound is designed to take maximum advantage of real-time audio processing and to encourage interactive experiments.

MUSE

Overview

The primary objective in designing MUSE was to create a general purpose sonification tool, which can create musical and non-fatiguing sounds and still allow meaningful exploration of scientific data over extended periods of time. Since subjectivity is involved in mapping data to sound parameters (as is true in the case of color as well), interactive and flexible mappings from data to different sound parameters was an important goal.

MUSE is the product of this conception, created by a team of computer scientists and musicians working together. Although music composition skills have been central to the design of MUSE, the design of both the user interface and the data mappings are meant for a scientific audience. Often the final decisions in musical compositions were made after experimentation with scientific data sets.

In order to make this work accessible to a broader community, we have simplified the presentation of musical composition techniques used in this work. Nevertheless, we refer the readers to Cooper [6] for excellent yet concise definitions of musical terminology.

MUSE is written in C++ for the SGI platform and uses CSound for the specification of sounds to be generated. It works with the SGI audio chip. Almost all newer SGI machines contain an audio chip so that sonification can be done without adding any additional device.

MUSE has two main components -- the music composition component, and the graphical user interface. The main objective of the music composition component is to specify sound that can be sent to CSound. Most of the rest of this section is devoted to describing how we achieve this.

The main objective of the graphical user interface is to provide an interactive and flexible environment for the user to map data to sound parameters. The graphical user interface of MUSE consists of menus, sliders, and buttons. The interface has been created using OpenGL and XFORMS.

Sound Mappings

Currently MUSE supports six different types of mappings of data to sound parameters. These parameters are: timbre, rhythm, tempo, volume, pitch (melody) and harmony.

Timbre: Data can be mapped to two different types of timbres -- instrument types and voice types. Currently supported instruments are strings, brass, oboe, clarinet, piano, and flute. Voice type timbres are supported by MUSE in the form of different vowels -- a, e, i, o, u -- in the human voice.

Rhythm: Metrical rhythm is a pattern of accents or stresses occurring regularly in time. Currently MUSE supports seven different types of rhythms -- flat, bouree, sarabande, minuet, waltz, swing, and gigue. Stress patterns in each of these rhythms can be completely specified by giving (i) start time, (ii) duration, and (iii) relative amplitude of all the notes in the rhythm.

Tempo: Tempo can be referred to as the speed at which the music is heard. In MUSE, data can be mapped to tempo. As data values increase, the tempo slows down and vice-versa. The reverse mapping can also be used.

Volume: In MUSE, data can also be mapped to volume. As data values increases, the volume increases and vice-versa. The reverse mapping can also be used.

Pitch (melody): MUSE supports mapping of data to pitches. Melody, in fact, is nothing but a succession of pitches. Mapping data to melody is perhaps the most difficult and challenging task. To meet this challenge, we have used the following principles of melodic construction [7, 12]. First, it is appropriate to choose a scale. That is, one should use only a certain subset of pitches within an octave. Although there are many different scales, in MUSE, we have currently

chosen the major scale because it is the most common scale in western music. The major scale consists of eight preselected pitches within an octave. Second, we describe the principle that is effective in bringing out the data meanings while preserving the character of melody. This principal requires having primarily stepwise motion between two successive pitches within an octave. In other words, there should not be a large gap between two pitches. In order to implement this principle, we have created interpolation patterns that quickly progress from one pitch to another in the case where there is a large gap between two successive pitches. This feature allows ears to estimate the pitch changes more effectively than if the sound were to jump directly from a lower pitch to a much higher pitch or vice-versa.

MUSE stores these interpolation patterns in a separate file and uses these patterns to generate score files dynamically at run time.

Harmony: Harmony is the resultant of the simultaneous combination of two or more musical sounds. In MUSE, data can be mapped to harmony so that higher data values (higher uncertainty) get mapped to more dissonant harmonies and lower data values (lower uncertainty) gets mapped to consonant harmonies. We have so far supported four distinct levels of harmonies that an untrained user can conveniently detect by ear. Each level of harmony in MUSE consists of four pitches. To construct the first two levels of harmonies, the principle of the harmonic series is used [6]. The harmonic series is the series of overtones that exist over any fundamental pitch. The higher up the harmonic series, the more dissonant the harmonies become. The first level of harmony uses four pitches, which are 1, 2, 3 and 4 times the original pitch, while the second level of harmony uses four pitches, which are 1, 5, 6, and 7 times the original pitch. The final two levels of harmonies are created using the theory of triad of pitches and tone clusters respectively [6].

Applications

We have applied MUSE to visualize uncertainty in isosurfaces and volumetric data. We have developed a system for visualizing uncertainty or differences in isosurfaces arising due to the use of different models such as trilinear interpolation for rectilinear grids, global multiquadrics, volume splines, and local multiquadrics for scattered data. Uncertainty is being visualized using a variety of techniques including pseudo-coloring, overlays, differencing, transparency, glyphs, and animation. In conjunction with visualization of uncertainty, differences in values of isosurfaces and volumetric data have been mapped to different parameters of musical sound -- pitch (melody), volume, tempo, timbre, harmony and rhythm. Although initial experiments are encouraging, further work is needed to establish scientific value of musical mappings, both by enriching the world of musical sounds created by the system and by evaluating the performance of human users against specific tasks.

CONCLUSIONS AND FUTURE WORK

We have presented MUSE -- a system for exploring scientific data using musical sonification. MUSE has opened up many possibilities for exciting further research. First, MUSE can be extended by incorporating many different types of instruments, rhythms, melodies, interpolation patterns, and scales, to name a few. Another possibility is to incorporate unpitched familiar sounds such as the sound of dragging (that can be associated with friction) in the sonification system. We plan to pursue this exciting direction of research in the future.

ACKNOWLEDGEMENTS

This work was partially supported by National Science Foundation grants IRI-9423881 and CDA-9115268, and by the faculty research funds granted by the University of California, Santa Cruz. We also want to thank David Marsh for assisting us in preparing a final version of this paper.

REFERENCES

1. U. Axen and I. Choi. Investigating geometric data with sound. In *Proceedings of the Third International Conference on Auditory Display, Palo Alto*, pages 25--28. 1996.
2. P. Astheimer. Sonification tools to supplement dataflow visualization. In Patrizia Palamidese, editor, *Scientific Visualization: Advanced Software Techniques*, pages 15--36. Ellis Horwood, 1993.
3. Stephen Barrass. Personify: a toolkit for perceptually meaningful sonification. *ACMA '95*, 1995.
4. R. Brady, R. Bargar, I. Choi, and J. Reitzer. Auditory bread crumbs for navigating volumetric data. In *Proceedings of the Late Breaking Hot Topics of IEEE Visualization '96*, pages 25--27. IEEE Computer Society Press, 1996.
5. R. Bargar, I. Choi, S. Das, and C. Goudeseune. Model-based interactive sound for an immersive virtual environment. In *Proceedings of the International Computer Music Conference, Tokyo, Japan*. International Computer Music Association, 1994.
6. Paul Cooper. *Perspectives in Music Theory: An Historical-Analytical Approach*. Harper & Row, New York, 1973.
7. W. Jay Dowling. Melodic contour in hearing and remembering melodies. In R. Aiello and J. A. Slobada, editors, *Musical Perceptions*, pages 173--190. Oxford University Press, 1994.
8. Steven P. Frysinger. Applied research in auditory data representation. *Proceedings of SPIE '90 Conference on Extracting Meaning from Complex Data*, 1259:130--139, 1990.

9. John P. Gather. Amsterdam catalogue of CSound computer instruments. http://mars.let.uva.nl/gather/accci, 1995.
10. W. W. Gaver. Synthesizing auditory icons. In *Proceedings of INTERCHI*, pages 228--235. 1993.
11. G.G. Grinstein and R.M. Pickett. EXVIS - an exploratory visualization environment. *Proceedings of Graphics Interface '89*, 1989.
12. Paul Hindemith. *The Craft of Musical Composition, Book II*. Associated Music Publishers Inc., New York, 1941. Translated by Otto Ortmann.
13. G. Kramer. *Auditory Display, Sonification, Audification, and Auditory Interfaces*. Addison-Wesley, 1994.
14. G. Kramer. An introduction to auditory display. In Gregory Kramer, editor, *Auditory Display, Sonification, Audification, and Auditory Interfaces*, pages 1--78. Addison-Wesley, 1994.
15. S. K. Lodha, C. M. Wilson, and R. E. Sheehan. LISTEN: sounding uncertainty visualization. In *Proceedings of Visualization 96*. IEEE, 1996.
16. R. Minghim and A.R. Forrest. An illustrated analysis of sonification for scientific visualization. In *Proceedings of Visualization 95*, pages 110--117. IEEE, 1995.
17. Tara Madhyastha and Daniel Reed. Data sonification: Do you hear what I see? *IEEE Software*, 12(2):85--90, March 1995.
18. Carla Scaletti. Sound synthesis algorithms for auditory data representations. In Gregory Kramer, editor, *Auditory Display, Sonification, Audification, and Auditory Interfaces*, pages 223--252. Addison-Wesley, 1994.
19. Barry Vercoe. *CSound Manual*. MIT, 1986. http://www.leeds.ac.uk/music/Man/Csound/contents.html.
20. C. M. Wilson and S. K. Lodha. LISTEN: a data sonification toolkit. In *Proceedings of the Third International Conference on Auditory Display, Palo Alto*, pages 35--40. 1996.