After Direct Manipulation - Direct Sonification

Mikael Fernström, Caolan McNamara

Interaction Design Centre, University of Limerick
Ireland

Abstract

The effectiveness of providing multiple-stream audio to support browsing on a computer was investigated through the iterative development and evaluation of a series of *sonic browser* prototypes. The data set used was a database containing music. Interactive sonification[1] was provided in conjunction with simplified human-computer interaction sequences. It was investigated to what extent interactive sonification with multiple-stream audio could enhance browsing tasks, compared to interactive sonification with single-stream audio support. With ten users it was found that with interactive multiple-stream audio the users could accurately complete the browsing tasks significantly faster than those who had single-stream audio support.

1 Introduction

This paper concerns the development of an interface that allows musicologists to browse musical data sets in novel ways. The data set (in the users' language often called a *collection*) is used by musicologists in their research. It contains over 7,000 tunes where each tune is represented by its score and a number of properties such as tonality, structure, etc. The traditional format for a collection is a printed book with various indexes. A common problem that musicologists have to deal with is to determine if tunes they collect in their field work exist in a particular collection, and if so, how they are related to other tunes in the collection, e.g. in chronology, typology.

1.1 Browsing

Browsing has become a popular term in recent years with the emergence of hypertext systems and the World Wide Web, but the concept of browsing goes well beyond these fields of application. There are many ways integrating text, sound, images and video to provide richer and more interesting systems that would allow us to use more of our natural abilities. Marchionini and Shneiderman [1] defined browsing as:

- "an exploratory, information seeking strategy that depends upon serendipity"
- "especially appropriate for ill-defined problems and for exploring new task domains"

This is the case when musicologists are searching for tunes in a collection. Tunes collected through fieldwork can often be different to older original versions. They can still be the same tunes but with the addition of an individual performer's style. This makes it difficult to use normal computer-based search algorithms [2]. Humans have an outstanding ability to recognise similarities in this domain, which suggests that in a good solution we should make use of our auditory abilities. However, current interfaces do not provide much in the way of support for browsing sounds.

1.2 Direct Manipulation

Most of today's interactive multimedia applications use direct manipulation. Items on display that can be interacted with can for example be highlighted when the cursor is over them, or the shape of the cursor can change. When an object is selected by a single mouse-button click, the object shows that it has been selected and when double-clicked the functionality associated with the object is activated.
We can summarise direct manipulation in the following words [3, 4] :

- Continuous representation of the objects of interest.
- Physical actions instead of complex syntax
- Rapid incremental reversible operations whose impact on the object of interest is immediately visible.
- Users get immediate feedback from their actions.

1.2 Browsing with sound support

In everyday listening one is often exposed to hundreds of different sounds simultaneously and is still able to pick out important parts of the auditory scene. With musical sounds, or tunes, many different factors affect our ability to differentiate and select between the sources. Using instrumental sounds, the timbre, envelope, tonal range and spatial cues support the formation of *auditory streams* [5, pp 455-528]. The tunes themselves also assist the formation of streams, as music has its own inherent syntactic and semantic properties [6]. It is also interesting to note the "cocktail party" effect, i.e. that it is possible to switch one's attention at will between sounds or tunes [7-10, 11, p.103].

Albers [12, 13] added sounds to a web browser, but kept the use of sound at a fairly low level of interactivity. Various 'clicks' were used when users clicked *soft buttons* and selected menus. To indicate system events such as data transfer, launch of 'plug-ins' and for errors he used 'pops and clicks', sliding sounds and breaking of glass sounds. For feedback about content, various auditory icons were used to indicate what kind of file a hyperlink was pointing to and the file size of the content indicated by piano notes (activated when the cursor was on a hyperlink). He also created hybrid systems using combinations of auditory icons, auralisation and sound spatialisation to enhance operator performance in mission control work settings [14, 15].

LoPresti & Harris' *loudSPIRE* system [16] added auditory display to a visualisation system. This system is an interesting hybrid as it used three different layers for sonification. System events were represented by electronic-sounding tones associated with computers; data set objects were represented by percussive or atonal auditory icons parameterized for object properties; domain attributes were represented by themes of orchestral music, harmonious tonal sounds, parameterized for attribute value of a region.

Begault [17] demonstrated the use of 3-D sound spatialisation for use in cockpits and mission control, in order to enhance speech perception. Kobayashi and Schmandt [18] showed that multiple-stream speech perception can be enhanced through 3-D sound spatialisation, including the existence of a spatial/temporal relation for recall of position within a sample of streamed speech, i.e. that the auditory content can be mapped to spatial memory.

With multiple auditory streams it is interesting to note the problem with differences in the individual ability to differentiate between multiple sound sources. A metaphor for a user controllable function that makes it visible to the user is the application of an *aura* [19]. An aura, in this context, is a function that indicates the user's range of interest in a domain. The aura is the receiver of information in the domain.

2 Prototype Development

Three design iterations were performed. In the first iteration exploratory interviews with potential users were conducted. Mock-ups and use *scenarios*[2] were created together with potential users. In the second iteration, a prototype was created in a high-level authoring package and informally tested through subjective evaluation. In the third iteration, a prototype was developed in MS Visual C++. This prototype was then tested in a *Thinking-aloud* study [20].

2.1 Users, Tasks and Environment

Throughout the development process, groups of users participated in the design and evaluation. They were all musicologists familiar with traditional methods and resources for their work, i.e. fieldwork and access to collections of music in paper-based formats. All users had some experience with computers, for example word processing and email.

A task list was developed for the testing of the final prototype based on a scenario developed in the first iteration. The task list was considered realistic by users from the first and second iteration. The idea behind the task list was to get users to work primarily in the auditory modality, with the visual modality as additional support. A total of 13 tasks were created, of which three were visual. The reason for having three visual tasks was to make the overall session more realistic, and to break the 'monotony' that otherwise might develop. The order of tasks was randomly allocated to each user.

A workstation was set up in an open plan office which was similar to the normal work setting of the users. One PC[3] was running the *sonic browser*, another PC was used by the experimenter playing sample tunes that the user should try to locate by browsing in the *sonic browser*. The users' speech and actions were recorded on video.

2.5 Sonic Software

Normal multimedia PC's cannot play multiple sound files concurrently. This would, of course, prohibit the desired development. To work around this problem, new intermediate drivers for the sound devices were developed. The problem with existing drivers is that when a sound is to be played, the operating system allocates the physical sound device exclusively. To solve this problem, the intermediate drivers have to read sound files and transform them into a common output format. Sound spatialisation was implemented to assist the users in differentiating and locating tunes. With sampled sounds, 3-D spatialisation can be used, but currently there is no existing support for 3-D spatialisation of MIDI synthesizer sounds on PC sound cards. Only stereophonic *"pan"* with difference in loudness between the left and right channel is available on standard sound cards [21, 22]. The problem with different speeds and formats of source files applies to both sound files (such as WAV) and sound controlling files (such as MIDI). As the users had expressed a preference for melody lines with MIDI controlled synthesizer sounds, all further implementation work focused on stereophonic spatialisation with only the difference in loudness between the left and right channel as a cue for auditory spatial location.

The users found that they sometimes wanted the aura on, sometimes off, as this allowed them to shift their focus between the neighbourhood of tunes to finer differentiation between just a few tunes. The number of tunes within the aura can vary due to the location of the cursor in relation to the density of the data set. Therefore an on-off function was added and the radius of the aura was made user controllable.

3 Prototype Evaluation

The prototype was evaluated to test the hypothesis that the application of multiple-stream sound enhances browsing tasks, compared to browsing with single-stream sound. It was evaluated by ten musicologists divided into two groups. The first group had the aura function disabled, i.e. they only had one tune played at a time when the cursor was positioned on a tune object. The second group had the aura function enabled and they could switch the aura on or off or resize it at any time during the tests. With the aura on, this group could get up to 16 simultaneous auditory streams.

4 Results

In each specific task, the users were allowed to move the cursor around freely in the display and soundscape trying to find the target tune (the sample tune presented to them at the start of each task). Occasionally the single-stream tasks were solved faster than multiple-stream tasks, but in no case were these differences statistically significant. Cumulative times were significantly faster in the multiple stream condition ($p < .05$). Overall, for the ten auditory tasks, the total time it took the users to find the target tunes show that *all* users with multiple-stream sound support were faster than the users with single sound support. It was verified that there was no correlation between the users familiarity with computers and the task completion times. The users with multiple-stream sound support were approximately 27% faster at locating the target tunes.

From the Thinking Aloud study there is a good indication that users remember where they heard a tune before, since users that browse with the *aura* on hear more tunes. This indication is also supported by for example Kobayashi and Schmandt [18].

5. Discussion

There are many limits to the traditional forms of data sets. The paper-based version is merely a well-structured repository of information, but requires a substantial amount of skilled work (by the end-user) to be usable. A straight, text-based database version only slightly improves the accessibility. It takes substantial effort to compare a 'fuzzy' sample (target tune) to hundreds of possible score representations.

The interfaces in many standard applications from some of the larger software developers have become overloaded and complicated in the interaction sequences. Through a simplified interaction sequence, users can work efficiently and with a high degree of satisfaction. The results also show that through tight coupling of the interaction, we can create a more engaging interface. By shifting some of the load from the visual to the auditory modality, we can perceive more information and make better use of our natural ability to recognise complex and 'fuzzy' patterns through seeing *and* hearing.

6. Conclusion

We could add a new word *audibility* to Don Norman's [23] two key principles for good interaction design: *visibility* and *affordance*, because we are dealing with multimedia systems and sound in particular. Audibility, in this sense, is the concept of how well a system can use auditory representation in the human-computer interaction. If the audibility is good, the users will perform their work better, faster, with fewer errors, and a higher degree of satisfaction. If the use of sound in the user interface can provide more affordances, or affordances that are complementary to the visual interface, we have a system with good audibility.

This is also important for users with different abilities. By using sonic representations (or auditory display) in the human-computer interaction, the resulting applications will potentially be usable to visually impaired people.

There are many issues that need to be further investigated if we want to develop guidelines and tool-kits for good *audibility*. Further investigations in perception and cognition at high levels of environmental complexity are required. Many guidelines are based on extremely isolated experiments, hence it is difficult to apply such guidelines in real work settings. To get more realistic models for what we, as human beings, can process, combinations of seeing, hearing and interaction should be studied.

References

1. G. Marchionini and B. Shneiderman, “Finding facts versus browsing knowledge in hypertext systems,” *IEEE Computer*, vol. 19, pp. 70-80, 1988.

2. D. S. Ó Maidín, “A Programmer’s Environment for Music Analysis,” in *Department of Music*. Cork, Ireland: University College Cork, 1995, pp. 283.

3. E. L. Hutchins, J. D. Hollan, and D. Norman, “Direct manipulation interfaces,” in *User-Centred System Design*, D. Norman and S. draper, Eds. Hillsdale, NJ, USA: Lawrence Erlbaum Associates, 1986, pp. 87-124.

4. B. Shneiderman, “Direct Manipulation: A step beyond programming languages,” *IEEE, Computer*, vol. 16, pp. 57-69, 1983.

5. A. S. Bregman, *Auditory Scene Analysis: The Perceptual Organization of Sound.* Cambridge, MA, USA: M.I.T. Press, 1990.

6. M. L. Serafine, *Music as cognition : the development of thought in sound*. New York, USA: Columbia University Press, 1988.

7. B. Arons, “A Review of the Cocktail Party Effect,” *Journal of the American Voice I/O Society*, vol. 12, pp. 35-50, 1992.

8. E. C. Cherry, “Some Experiments on the Recognition of Speech with One and Two Ears,” *Journal of the Acoustical Society of America*, vol. 25, pp. 975-979, 1953.

9. E. C. Cherry and W. K. Taylor, “Some Further Experiments on the Recognition of Speech with One and Two Ears,” *Journal of the Acoustical Society of America*, vol. 26, pp. 549-554, 1954.

10. C. Schmandt and D. Roy, “Using Acoustic Structure in a Hand-held Audio Playback Device,” *IBM Systems Journal*, vol. 35, 1996.

11. C. D. Wickens, *Engineering Psychology and Human Performance*. NY, USA: Harper-Collins Publ. Inc, 1992.

12. M. Albers and A. S. Bergman, “The Audible Web: Auditory Enhancements for Mosaic,” Proceedings of ACM CHI ’95, Denver, CO, USA, 1995.

13. M. C. Albers, “Auditory Cues for Browsing, Surfing and Navigating,” Proceedings of ICAD ’96, Palo Alto, CA, USA, 1996.

14. M. Albers, “Sonification and Auditory Icons in a Complex, Supervisory Control Environment,” Proceedings of ACM SIGGRAPH/Multimedia `93, Workshop, Sound-Related Computation, Los Angeles, CA, USA, 1993.

15. M. C. Albers, “Varese - Non-speech Autitory Interfaces,” in *Industrial and Systems Engineering*. Atlanta, Georgia, USA: Georgia Institute of Technology, 1995.

16. E. LoPresti and W. M. Harris, “loudSPIRE, and Auditory Display Scema for the SPIRE System,” Proceedings of ICAD ’96, Palo Alto, CA, USA, 1996.

17. D. R. Begault, *3-D Sound for Virtual Reality and Multimedia.* Cambridge, MA, USA: Academic Press Inc, 1994.

18. M. Kobayashi and C. Schmandt, “Dynamic Soundscape: mapping time to space for audio browsing,” Proceedings of CHI’97, Atlanta, GA, USA, 1997.

19. S. Benford and C. Greenhalgh, “Introducting Third Party Objects into the Spatial Model of Interaction.,” Proceedings of BCS HCI ’97, Bristol, UK, 1997.

20. J. Gould, “How to design usable systems,” in *Handbook of Human-Computer Interaction*, M. Helander, Ed. Amsterdam, Holland: North-Holland, 1988, pp. pp. 757-790.

21. CreativeLabs, “Part V Audio Spatialization Library API,” Creative Labs, Inc., Developer’s Information Pack 1996.

22. Microsoft, “DirectX 2 SDK,” Microsoft Corporation, Developers Kit 1996.

23. D. A. Norman, *The Psychology of Everyday Things*. NY, USA: Basic Books Inc., 1988.

[1] The term *sonification* was chosen for several reasons. Although, in this particular scenario, musical tunes were used, timbre and auditory spatial location was used to enhance segregation. In other experiments with our *Sonic Browser*, data sets with for example environmental pollution data has been used, hence in that case the underlying objects could be considered to be *earcons*.

[2] describing the context of use, the users and their work and environment.

[3] Intel Pentium, 120 MHz, 32 MB RAM, 17" display with 1024 x 768 pixels in 16-bit colour, Creative Labs SoundBlaster 16 sound card with OPL3 FM synthesis, loudspeakers Altec Lansing 'Multimedia' stereo 2 x 5 W, Microsoft Windows 95 v4.