

ENHANCING SONIC BROWSING USING AUDIO INFORMATION RETRIEVAL

Eoin Brazil, Mikael Fernström

George Tzanetakis, Perry Cook

Interaction Design Centre

Department of Computer Science and
Information Systems
University of Limerick, Ireland

eoin.brazil@ul.ie

Computer Science Department
Princeton University
USA

gtzan@cs.princeton.edu

ABSTRACT

Collections of sound and music of increasing size and diversity are used both by typical computer users and multimedia designers. Browsing audio collections poses several challenges to the design of effective user interfaces. Recent techniques in audio information retrieval allow the automatic extraction of audio content information. This information can be used to inform and enhance audio browsing tools. In this paper we describe how audio information retrieval can be utilized to create novel user interfaces for browsing of audio collections. More specifically we report on recent work on two system prototypes: the *Sonic Browser* and *Marsyas* and our current work on merging the two systems in a common flexible system.

1. INTRODUCTION

Multimedia browsing and searching is a hard problem. The information is diverse and challenges even confounds any single visualization or classification technique. In our opinion, only a collection of complementary techniques will truly suffice, with the strength of one compensating for the weakness of another. Our goal is to explore and combine existing techniques to reveal new solutions to the problem of browsing and searching large sound collections.

The amount of audio information and resources available on personal computers and via the Internet has grown exponentially over the past decade. This is referred to as the data availability paradox by Woods [1], more and more data is available, but our ability to interpret what is available has not increased. In the context of this research, we are interested in facilitating human browsing behaviour. Browsing can be defined as “an exploratory, information seeking strategy that depends upon serendipity ... especially appropriate for ill-defined problems and for exploring new task domains” [2]. Although visual metaphors for browsing text files and images have been explored there is little work done on browsing audio collections. Most current sound editor packages and system tools are centred on the notion of editing or processing a single sound file at a time. The key issue addressed in this paper is the provision of mechanisms whereby a user can explore large collection of sounds. Our approach involves mapping the sound collection after analysis to a two- or two & half or three-dimensional representation to create novel graphical user interfaces. With a purely visual representation, it is difficult to give the user useful clues about which of the multiple related

sounds to select. This is avoided, as the space is both visually and aurally represented.

There has been much research into the area of image retrieval which has been spurred on by the now commonplace availability of digital photography to the home user. Thumbnail (miniature) pictures and various spatial techniques have shown improvements in browsing and retrieval of images [3]. Limited work has been done regarding tools exploiting derivatives of these techniques that can be applied to sounds.

In order to improve the user interaction with audio collections, novel User Interfaces that allow simultaneous presentation of multiple sound clips for browsing are required. In recent years several techniques for the automatic extraction of information from audio signals have been proposed. An early overview of audio information retrieval can be found in Foote [4]. Using these techniques, audio signals can be analysed in many different ways such as: retrieval based on similarity, classification to various types of audio, audio thumbnailing, and segmentation based on sound “texture”. In this paper we describe how these assist and enhance the functionality of audio browsing tools through the presentation of recent work on two prototype audio browsing systems: the *Sonic Browser* and *Marsyas* and our current work on merging them into a distributed, flexible, and powerful audio browsing system called the *Audio Retrieval Browser (ARB)*. The *Sonic Browser* offers the ability to browse several sound files simultaneously and to navigate through a stereo-spatialised soundscape, while *Marsyas* offers audio analysis, synthesis, genre classification, segmentation-based thumbnailing *Timbregrams* and *TimbreSpaces* (new audio collection visualizations) [5].

In both systems the central idea is to map sound clips to aural and/or visual objects with properties that convey information about the sound clips and use these objects in order to create browsing spaces. The ARB as well as the two previous systems have been designed with Shneiderman’s mantra for design of direct manipulation & interactive visualisation interfaces “**overview first, zoom and filter, then detail on demand**” in mind [6].

2. HISTORY AND RELATED WORK

In this paper we have narrowed sound browsing and classification to the areas of specific sound classification, extended musical genre classification, open-ended browsing

and the exploration of sound resources in large and complex data sets.

The basis for this research was the possible combination of best features from two different software systems designed for browsing sound resources: the *MARSYAS* framework for audio analysis [7] and the *Sonic Browser* application [8]. These two systems were presented at ICAD 2001, where the authors discovered the common goals and complementary approaches of the two projects and decided to combine them in one system. The merging of these systems aims to produce a graphical user interface which allows for flexible manual and automatic setting of application parameters providing a 2D or 2.5D soundscape with interactive aura playback of multiple stream audio. This application (*ARB-Audio Retrieval Browser*) provides an interface for browsing and interacting with large digital sound collections.

2.1. The projects

The **Sonic Browser** is a tool for accessing sounds or collections of sounds using sound spatialization and context-overview visualization techniques. The *Sonic Browser* uses multiple stream stereo-spatialised audio activated by cursor/aura-over-icorns, representing sound files. Using the *Sonic Browser*, properties of the sonic objects can be mapped to arbitrary features of the visual display. As an example, file size can be mapped to size of visual symbols, sampling rates to colour, symbol shape to file type and horizontal and vertical location to date and time. A metaphor for a user controllable function that makes it visible to the user is the aura [9]. An aura, in this context, is a function that defines the user's range of perception in a domain. The aura is the receiver of information in the domain, shown as a grey shaded circle surrounding the cursor in Figure 1, the cursor is surrounded by the aura. All sonic objects within the aura play simultaneously, panned out in a stereo-space around the cursor.

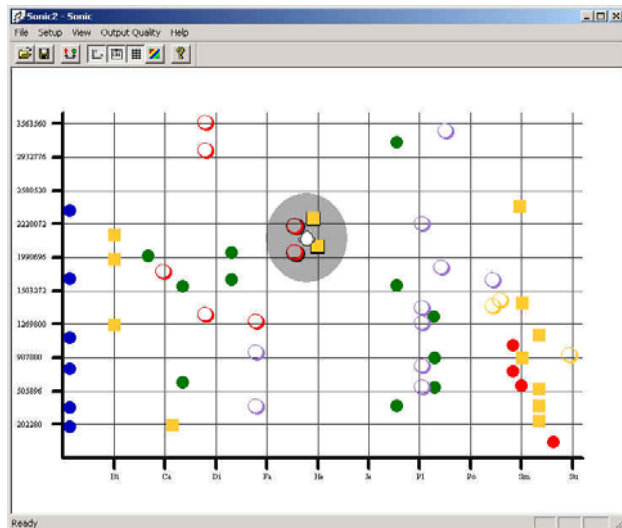


Figure 1: *Sonic Browser*

MARSYAS is a computer audition research software framework that allows users to manipulate, analyse and retrieve from large audio collections. A general overview of the system can be found at [7]. It supports various types of automatic audio feature extraction, beat analysis, classification, clustering, segmentation, content-based similarity retrieval, and

thumbnailing. Based on these techniques a variety of graphical user interfaces are supported. The *Enhanced Sound Editor* in addition to standard audio editing functionality (waveform and spectrogram displays, mouse selection, zooming etc) provides automatic classification, segmentation and retrieval. The *Timbregram Browser* represents sound files as *Timbregrams* [5], which are a static context, and content dependent visualization of audio and the *TimbreSpace* browser represents audio files as points in a 2.5D space. Figure 2 shows a screenshot of the system, showing 5 *Timbregrams* on the left, a *TimbreSpace* on the right, and one of the files opened in the *Enhanced Sound Editor* on the bottom. More information about these interfaces and their use with the Princeton Display Wall can be found at [10].

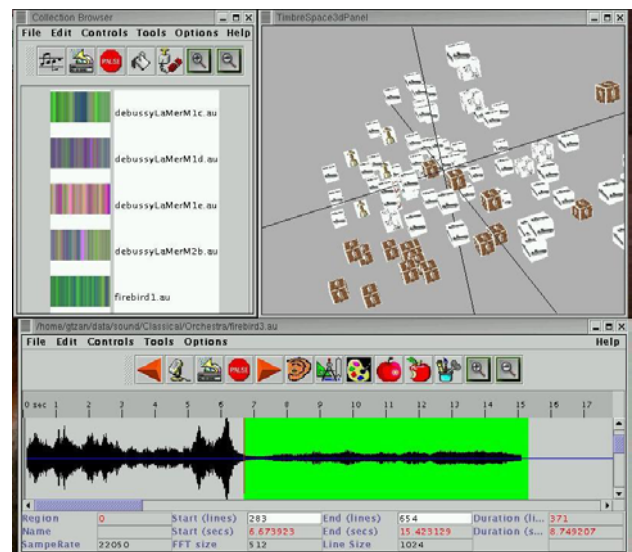


Figure 2: *MARSYAS*

3. AUDIO RETRIEVAL BROWSER

The **Audio Retrieval Browser** is a tool that combines the best features of both previously described projects and provides novel ways to browse large audio collections. More specifically it combines the interactivity and user configuration of the *Sonic Browser* with the automatic audio information retrieval of *MARSYAS* in order to create a flexible distributed graphical user interface with multiple visualization components that provides novel ways of browsing large audio collections.

3.1. Sound object, spaces and mappings

The central idea of the ARB is to represent sounds as visual and aural objects with specific properties. Browsing spaces consist of collections of these objects. For example sounds can be represented as colored shapes (objects) on a 2D plane (space). The properties of the sounds can be mapped to arbitrary features of the visual display. File size can be mapped to size of visual symbols, sampling rates to colour, symbol shape to file type and horizontal and vertical location to date and time. Mappings that should also be considered are simile, acoustic/perceptual, onomatopoeia and event. Simile is saying where one sound is like another sound or group of sounds. Acoustic/perceptual mappings describe a sound's physical attributes such as brightness, dullness or pitch. Onomatopoeia is where sounds are described by the way they sound, e.g.

hammering could be "thunk-thunk" [11]. Audio events / actions are associated with what action or event that is happening, for example, a sound of a car braking would be "braking".

Alternatively, the features of the visual interface can be mapped to represent the content of the sound files using automatic audio information retrieval techniques. For example the object shape or color can be automatically derived based on automatic musical genre classification and the horizontal location can be mapped to beat strength [12]. Using a mapping like that the user can easily locate a fast jazz piece in a large collection that contains various different types of music. As another example automatic clustering using the k-means algorithm on acoustic features can be used to group sound effects based on their similarity. Another possibility is the use of dimensionality reduction techniques such as Principal Component Analysis to map high dimensional feature vectors to color or coordinates (see Timbregrams, Timbrespaces in [7]). Although a detailed presentation of the types of automatic audio information retrieval that can be used to create mappings is beyond of the scope of this paper we mention a few representative examples that are supported by our system: Features (Fast Fourier Transform, Pitch, Beat, Beat Strength, Mel-Frequency Cepstral Coefficients, Linear Prediction Coefficients), Classification (Gaussian, Gaussian Mixture Model, K-Nearest Neighbors, BackPropagation Neural Network), Clustering (K-Means, Learning Vector Quantization). Essentially the foundation of most audio information retrieval algorithms is the representation of the signal as a series of numbers called the feature vector. For example in automatic musical genre classification [12] information about the short-time timbral texture, rhythmic structure, and harmonic content is utilized. In most cases in order to calculate the feature vector some technique of analyzing the signal in time and frequency, such as the Short Time Fourier Transform or Wavelets, is used. Once sound files are represented as feature vectors, standard pattern recognition [13] techniques can be used to train statistical models of the feature vectors that can be used for audio information retrieval. Training can be supervised or unsupervised. More information about how these techniques can be applied to audio signals can be found in [7].

The users can at any time change these arbitrary mappings, to suit their needs. This allows users to determine a mapping suitable for their sound objects and the related sound collection something which is important given the subjective nature of how humans perceive sounds which is impossible to fully capture automatically. The combination of automatically derived mappings with manual editing allows a semi-automatic approach to the creation of browsing spaces that is not as time-consuming as a complete manual configuration while at the same time retaining the flexibility of the manual approach.

With tight coupling between the visual and auditory information, users rapidly get a good spatial idea of what objects that are available and how to navigate between them. This meets Shneiderman's requirements for rapid, incremental, and reversible queries while still offering a continuous display and progressive refinement of the query. Exploiting this tight coupling a user can attend to more information per unit time. Information processing capacity is increased, thereby amplifying cognition. This guiding principle is explicitly stated as the principle of reducing the cost structure of information [14].

The visual parameters within the display are based on Bertin's non-spatial attributes [15]: color, shape, size, saturation, texture, and orientation. Spatial attributes such as 2D

or 3D position are dependant upon the arbitrary mapping. The typical configuration of the display is dependent upon the mapping. These mappings are automatically derived from the classification techniques mentioned previously. The addition of a "tagging" mechanism for objects of interest has also been noted in earlier research [8] and in the ARB a simple shading of objects is implemented.

The aura facilitates our ability to switch our attention between different sounds in the auditory scene, making use of the "cocktail party effect" [16]. The aura is user controlled and can be reduced or increased in size, by reducing the aura a user can "zoom in" on a particular sound/s. A sound under the aura can also be dragged using a "hold & drag" mouse operation. This allows for the user to manually reclassify the sound within the collection. The objects under the aura can provide "detail on demand" by a simple mouse click that displays the particular sound/s properties, both aurally and visually.

3.2. Visualization components

Information visualisation techniques offer many components, which can be applied to a limited display surface used in accessing large volumes of data. There are two distinct categories: distortion-oriented and nondistortion-oriented [17]. The ARB uses mechanisms from both categories. The techniques used are mainly from the branch of distortion-oriented techniques known as focus + context. Analyses of these techniques have been written by Noik [18] and by Leung [17], which we will not duplicate here. Two-dimensional techniques which have already been used in our applications included x, y plots and Treemaps [19, 20].

Two & half and three- dimensional techniques which have been used in the ARB include mapping audio files to a 3-dimensional space. Principal Component Analysis [21] can be used to reduce the dimensionality of the feature vector representing the file to the 3-dimensional feature vector corresponding to the point coordinates. Coloring of the points is based on the automatic genre classification.

In addition to the above-mentioned techniques the ARB supports two other techniques that were not supported in the previous systems. The first technique is the hyperbolic tree [22] is a well-known focus + context technique, it is illustrated in Figure 4. A tree is kept within the confines of a circular area on the screen; this method is based on a hyperbolic geometric transformation. This idea stems from where the whole circle is the aura, but objects, which reside on the perimeter are of the circle silent, while objects in the center are played. A central object is played full volume and full stereo while an object to the left of the centre is played panned to the left and the relative loudness is in inverse square relation to the centre of the circle / aura. The second technique from the area of filtering and visualization is modified from two previous techniques, the Alphaslider [23] and the Lensbar [24]. Three sliders which are related to specific sound attributes, sound source, audio event / action and onomatopoeia with the sliders augmented by use of a filter and zooming mechanism.

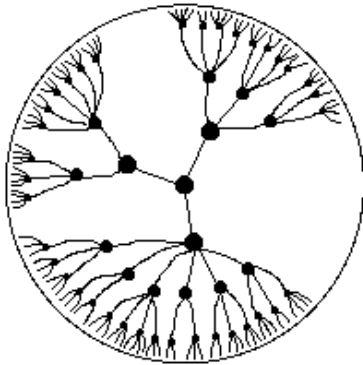


Figure 4: A Hyperbolic Tree mapped onto a circle

Supporting multiple visualizations provides the necessary flexibility to support a variety of different applications. The separation of the mapping of sound properties to display configuration from the visualization components provides additional flexibility. For example, a better algorithm for sound classification can easily be incorporated into the system and potentially be used by any visualization component without code rewriting.

4. ARCHITECTURE-IMPLEMENTATION

The ARB is designed to be a distributed system with multiple components rather than a single monolithic integrated application. Communication between the components is achieved using the Extensible Markup Language (XML). Using this approach the system is flexible and new components or changes can be integrated with minimal effort. Moreover, most of the existing code base of the two previous projects was retained in the new system.

4.1. Extensible Markup Language - XML

XML (Extensible Markup Language) is a markup language defined by the World Wide Web Consortium. It allows for the creation of user defined markup languages for storing structured data in text files. The ARB uses XML in two areas: configuration and sound meta-data. Configuration data contains information on the database setup, server setup and GUI preferences & layout.

Information on the visual properties of the sound within the GUI is also dynamically configurable using XML. Visual properties within the soundscape include x, y, z coordinates, colour and shape. Initially these properties are automatically configured but the users may recategorise any element within the dataset from the GUI. This allows for the manual dynamic setting of these properties. The sound meta data is stored in a similar fashion and holds information similar to ID3 tags in MP3 files [25, 26] and Cue Points [27] depending on the file format.

4.2. Database Implementation

The database is implemented using the MySQL database system, which communicates with the application using the JDBC protocol. Using the information stored in the database, it is possible for users to recreate their collections from different machines. It also stores the procedures used by a user to classify a sound with a visualisation aspect.

4.3. Client Server Architecture

The Client-Server architecture was chosen for the application as it allows the client to be freed from complex calculations, which take place on the server. The server acts as a distributed repository for user's sound resources. Sound collections can be on the same computer, or in the same LAN or anywhere a user's computer can connect to the server with sufficient bandwidth. The issues of copyright control and rights management have also been left unaddressed in this research, as it is orthogonal to our goals for this system.

4.4. Implementation

The graphical user interfaces are written in JAVA and the audio information retrieval engine is written in C++ for faster numerical processing. The C++ code is portable and has compiled in Linux, Solaris, Irix, Windows (Visual Studio and Cygwin compilers). The choice of JAVA, C++ and XML for the communication layer makes the system portable to new architectures. The client-server architecture allows a variety of configurations. For example, it is possible to have Windows and Solaris clients accessing the audio information retrieval engine running in Linux. A screenshot of the ARB using the TreeMap visualization is shown below in Figure 5 and using the HyperTree visualisation in Figure 6.

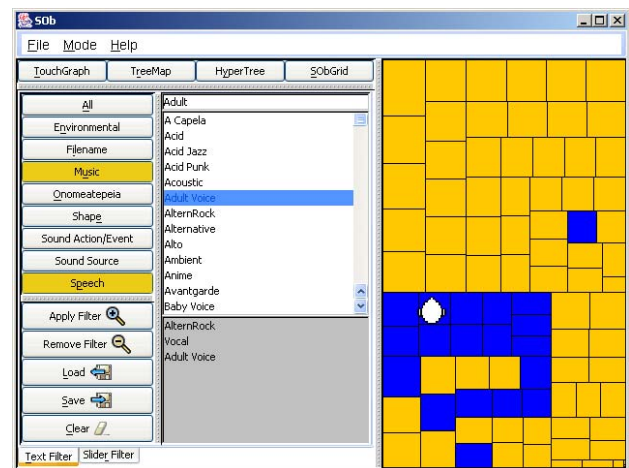


Figure 5: The ARB with the TreeMap visualization

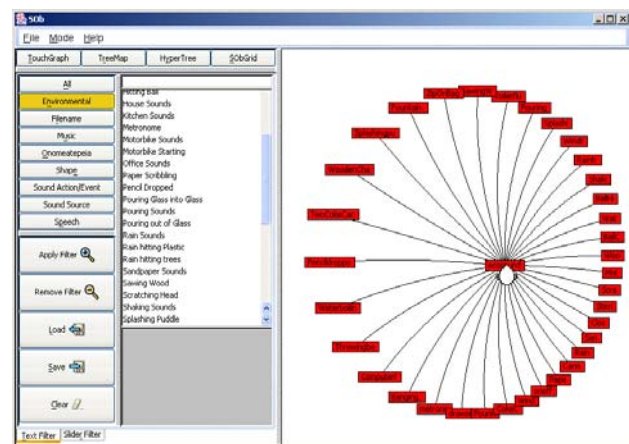


Figure 6: The ARB with the HyperTree visualization

5. SCENARIOS

The system is being investigated using several scenarios and their related audio collections. The first collection consists of various everyday sounds taken from a library of digital sound effects and the second consists of various pieces of music. Different automatic and manual mappings and visualization components will be demonstrated. The investigation of our tools has been driven by a number of scenarios, two of which are described here.

5.1. Psychoacoustic Experimental Tool

The first scenario uses the system as a psychoacoustic experiment tool. We are exploring this scenario using a sound collection of sound effects. This provides a collection for the user to navigate and then further refine the classification of these sounds. Here the particular requirement is for the further classification of the sound's properties according to a psychoacoustic scaling. The scaling used for this scenario represents a 'Gaver-scape' [28].

The ARB is not currently intended as a substitute for traditional MDS analyses; rather, its purpose in this scenario is to collect data using a format that increases participants' motivation by reducing perceptual and cognitive task demands and by encouraging innovative decisional strategies. Additional analysis using traditional designs can be performed on subsets of stimuli. The similarity estimates created by the users are stored in the system's database allowing the easy collection and processing of similarity data for a large number of sounds.

5.2. Sound Designer Tool

Our second scenario involves both collections. This scenario investigates the various classification and visualization techniques used by the ARB. The user interface allows users to select which audio retrieval techniques are used for the queries in the application. This allows the user to compare queries using the audio retrieval techniques and also the information visualization techniques. The results of these queries allow the user to better understand the effect of the both the retrieval and visualization technique on the query. As with the first scenario, users can further refine the classification of the sounds and this information can then be used to further train automatic classification systems.

The ARB offers several features that benefit users working with large sound collections. A Sound Designer could formulate a query, an example being a query for all the sounds with the properties of hitting, wailing, tinny, rain sounds. The resulting soundscape is then displayed and can be browsed with sounds of interest being tagged for future use. The query could be further refined or a new query created or the user can browse the entire collection.

6. RESULTS – DISCUSSION

Sound browsing and searching using new visualization and classification methods has enormous potential for managing large sound collections. The ARB shows evidence of providing a robust tool with different visualization methods for subtly distinguishing sounds based on distinct classifications.

We have a number of research aspects that we would like to expand upon to fully realise the ARB's potential. The ARB

system must undergo a rigorous benchmarking procedure, comparing its performance with other large-scale audio browsing tools. Our initial, informal results are encouraging. In addition, we are preparing to add a collaborative classification element to test ARB on music collections using peer classification to supplement the automatic analysis.

ARB's is currently limited by some external factors, principally the requirement to manually classify certain categories of sound such as Environmental or Sound Source. The manual classification of sounds is a very time consuming method, hindering the goal of a real time audio system. MARSYAS classification will aid in this area as it is actually faster than real time for the classification of sounds.

7. FURTHER RESEARCH

In the case of the ARB, there are a number of design improvements that make sense in light of our demonstration results.

An area of definite investigation is the use of inter-application operations such as drag and drop. This would allow a client to add a sound file to the collection and have it classified by the information retrieval engine. Future experiments should be carried out to further our understanding of how the design of interface elements and components of an interface facilitate the classification, searching and browsing of sound collections.

Another direction for future work is the use of a hyperbolic three-dimensional space technique such as the H3 layout technique defined by Munzner [29]. The XML3D technique [30], which is a more recent technique by Munzner, should also be investigated for suitability. Future investigation should examine these and any new research into the visualization of large scale datasets with a view to its implementation within the ARB.

Hierarchical clustering techniques [13] such as agglomerative clustering can be used to automatically derive tree structures for audio signals. We plan to explore this in the creation of hyperbolic trees.

Another possibility for future work is the use of the manual configuration of the interface to drive audio information retrieval algorithms. For example the user could manually mark file's visual properties based on some classification that is not supported by the system. The labelling information provided by the visual properties could be used to train a statistical pattern recognition classifier based on acoustic features to perform the same classification decision. This feature would be useful in conducting various user experiments about sound perception and directly feeding their results to train automatic systems.

The addition of a zooming mechanism for navigation of a clustered dataset is another area for research. As the size of the collections increase so does the cluster size of related sounds. Further research and user experiments into the "tagging" mechanism are required, to determine if simple "shading" of the object is the most effective mechanism for "tagging".

8. CONCLUSIONS

From the combination of MARSYAS and the *Sonic Browser* we have shown a graphical user interface which allows for flexible manual and automatic setting of application parameters providing a 2D or 3D soundscape with interactive aura playback of multiple stream audio. ARB provides an integrated multimodal user interface for browsing and classification of

large-scale sound datasets. The present prototype demonstrates the power of the system, while several additional functions remain to be implemented and tested, such as full zoom in/out and more algorithms for visualization. Informal user testing has indicated a high degree of satisfaction, while a more formal user testing remains to be done.

9. ACKNOWLEDGEMENTS

This work was supported by the European Union under the European Commission's Future & Emergent Technologies collaborative R&D programme (Sob – The Sounding Object Project – no IST-2000-25287; <http://www.soundobject.org>).

10. REFERENCES

- [1] D. D. Woods, E. S. Patterson, E. M. Roth, and K. Christoffersen, "Can we ever escape from data overload?," presented at 43rd Annual Meeting on Human Factors and Ergonomics Society, Houston, Texas, 1999.
- [2] G. Marchionini, *Information Seeking in Electronic Environments*. New York, USA: The Press Syndicate of the University of Cambridge, 1995.
- [3] K. Rodden, W. Basalaj, D. Sinclair, and K. Wood, "Does organisation by similarity assist image browsing?," presented at SIG-CHI on Human factors in computing systems, Seattle, WA USA, 2001.
- [4] J. Foote, "An overview of audio information retrieval," *ACM Multimedia Systems*, vol. 7, pp. 2-10, 1999.
- [5] G. Tzanetakis and P. Cook, "3D graphics tools for sound collections," presented at COST-G6 Conference on Digital Audio Effects DAFx-00, Verona, 2000.
- [6] B. Shneiderman, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 2 ed. Reading, MA, USA: Addison-Wesley, 1992.
- [7] G. Tzanetakis and P. Cook, "MARSYAS: a framework for audio analysis," *Organised Sound*, vol. 4, 2000.
- [8] J. M. Fernström and E. Brazil, "Sonic Browsing: An Auditory Tool For Multimedia Asset Management," presented at International Conference on Auditory Display ICAD-01, Espoo, Finland, 2001.
- [9] J. M. Fernström and C. McNamara, "After Direct Manipulation - Direct Sonification," presented at International Conference on Auditory Display ICAD '98, Glasgow, Scotland, 1998.
- [10] G. Tzanetakis and P. Cook, "MARSYAS3D: a prototype audio browser-editor using a large-scale immersive visual and audio display," presented at International Conference on Auditory Display ICAD-01, Espoo, Finland, 2001.
- [11] S. Wake and T. Asahi, "Sound Retrieval with Intuitive Verbal Expressions," presented at International Conference on Auditory Display ICAD-98, Glasgow, Scotland, 1998.
- [12] G. Tzanetakis, G. Essl, and P. Cook, "Automatic Musical Genre Classification Of Audio Signals," presented at International Symposium on Music Information Retrieval, Bloomington, Indiana, USA, 2001.
- [13] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd Edition ed. New York: Wiley, 2000.
- [14] S. K. Card, J. D. Mackinlay, and B. Schneiderman, *Information visualization: Using vision to think*. San Francisco: Morgan-Kaufmann, 1999.
- [15] J. Bertin, *Semiology of Graphics*. Milwaukee, Wis: University of Wisconsin Press, 1983.
- [16] B. Arons, "A Review of the Cocktail Party Effect," *Journal of the American Voice I/O Society*, vol. 12, pp. 35-50, 1992.
- [17] Y. K. Leung and M. D. Apperley, "A Review and Taxonomy of Distortion-Oriented Presentation Techniques," *ACM Transactions on Computer-Human Interaction*, vol. 1, pp. 126-160, 1994.
- [18] E. G. Noik, "Encoding presentation emphasis algorithms for graphs," presented at Graph Drawing '94, Lecture Notes in Computer Science, 1994.
- [19] B. Johnson and B. Shneiderman, "Tree-maps: A space filling approach to the visualization of hierarchical information structures," presented at IEEE Visualization '91, San Diego, CA, 1991.
- [20] B. Shneiderman, "Tree visualization with tree-maps: A 2-d space-filling approach," *ACM Transactions on Computer Graphics*, vol. 11, pp. 92-99, 1992.
- [21] L. Jolliffe, *Principal Component Analysis*: Springer Verlag, 1986.
- [22] J. Lamping, R. Rao, and P. Pirolli, "A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies," presented at CHI '95, Denver, CO, USA, 1995.
- [23] C. S. Ahlberg, B., "The alphaslider: a compact and rapid selector," presented at Human factors in computing systems, Boston, United States, 1994.
- [24] T. Masui, "LensBar - visualization for browsing and filtering large lists of data," presented at IEEE Symposium on Information Visualization, 1998.
- [25] M. Nilsson, "ID3v2 informal standard - Main Structure," in *Developer's Website*. Linköping: Open Source Project, 2000.
- [26] M. Nilsson, "ID3v2 informal standard - Native Frames," in *Developer's Website*. Linköping: Open Source Project, 2000.
- [27] E. Brazil, "Cue Points: An Introduction," presented at COST-G6 Conference on Digital Audio Effects DAFx-01, Limerick, Ireland, 2001.
- [28] W. W. Gaver, "What in the world do we hear? An ecological approach to auditory source perception," *Ecological Psychology*, vol. 5, pp. 1-29, 1993.
- [29] T. Munzner, "H3: Laying Out Large Directed Graphs in 3D Hyperbolic Space," presented at IEEE Symposium on Information Visualization, Phoenix, AZ, 1997.
- [30] T. Munzner, "Interactive Visualization of Large Graphs and Networks," in *Computer Science*: Stanford University, 2000.