AUDIOSCAPES: EXPLORING SURFACE INTERFACES FOR MUSIC EXPLORATION

Steven R. Ness, George Tzanetakis

University of Victoria
Department of Computer Science, Victoria, Canada
sness@sness.net,gtzan@cs.uvic.ca

ABSTRACT

There is a growing interest in touch-based and gestural interfaces as alternatives to the dominant mouse, keyboard and monitor interaction. Content and context-aware visualizations of audio collections have been proposed as a more effective way to interact with the increasing amounts of audio data available digitally. *Audioscapes* is a framework for prototyping and exploring how touch-based and gestural controllers can be used with state-of-the-art content and context-aware visualizations. By providing well-defined interfaces and conventions a variety of different audio collections, controllers and visualization methods can be combined to create innovative ways of interacting with large audio collections. We describe the overall system architecture, the currently available components and specific case studies.

1. INTRODUCTION

The size of digital audio collections has been steadily increasing due to a combination of factors including digital music distribution, increases in storage capacity, advances in audio compression and the wide popularity of portable digital music players. Effective interaction with these large audio collections poses significant challenges to traditional user interfaces. Portable players and music playlist management software typically allow users to select artists, genres or individual tracks by essentially browsing long lists of text. This mode of interaction, although adequate for small music collections, becomes increasingly problematic as the collections become larger. The emerging area of Music Information Retrieval (MIR) deals with all aspects of managing, analyzing and organizing music in digital formats. In the past ten years many MIR algorithms and user interfaces have been proposed that can assist with the browsing and navigation of large music collections.

Recently there has been an increasing interest in alternatives to the traditional mouse/keyboard human-computer interaction. Touch-based and gestural interfaces have changed status from research curiosities to being part of many mainstream consumer computing devices.

AudioScapes is a framework developed to explore the design space of non-traditional interfaces for audio and music collection browsing based on the metaphor of a surface. In this metaphor the individual audio recordings or music tracks are mapped onto a 2-dimensional surface which can be navigated using different controller interfaces. The overall abstract architecture captures the structure of the majority of existing systems and provides significant design flexibility in the choice of individual specific components. By providing well-defined interfaces and conventions, a variety of different audio collections, controllers and visualization methods can be combined to create innovative ways of interacting with large audio collections.

Our design goal is to provide effective interaction without relying on textual metadata. There are many usage scenarios where having to know artist names and album titles or having to read text is impractical. Currently the most common approach in these cases is just playing random songs (the so-called "shuffle"). Althout satisfactory for small and homogeneous collections this approach is not particularly effective for larger audio collections. These issues become even more pronounced when the users have vision and/or motion disabilities. For example, finding a particular artist out of a long list of text using a scroll-wheel can be very difficult or even impossible for a user with motor disabilities. Similarly, reading text on a screen is not directly possible for a blind user. We have used *AudioScapes* to design and prototype interfaces for such users. Although we do not focus on textual metadata, the proposed interfaces can be used in conjuction with standard text-based interfaces.

2. RELATED WORK

There has been considerable recent interest in the development of touch-based and gesture based interfaces[5]. This represents a movement from traditional Graphic User Interfaces (GUI) to Touch-Based User Interfaces (TUI) [2]. These new forms of interfaces help to bring together the virtual world with the real world, providing a more inclusive and immersive interaction environment for users. The iPhone is a device that supports multitouch interaction, a system where multiple fingers are tracked to provide different types of functionality. For example, a touch on the surface with one finger would produce a different effect than when three fingers are used. In addition gestures such as joining two fingers can be used for actions such as zooming.

In the field of Music Information Retrieval, data of high dimensionality and of considerable complexity is generated. Various visualization interfaces have been proposed to make this data accessible and useful to users. Frequently these interfaces rely on automatically extracted audio features. Islands of Music [9] is an example of such a visualization of audio information which uses the technique of Self-Organizing Maps to generate a two-dimensional representation of a collection of music. MusiCream [3] is an interface that allows users to interact with a music collection using a dynamic visualization interface. The Databionic/MusicMiner system [7] allows users to organize large collections of music and employs Emergent Self-Organizing Maps to generate visualizations of the data involved. A 2006 review of some of the recent trends in visualization in audio based music information retrieval can be found in Cooper [1].

Audioscapes is the evolution of several research efforts by our group [8, 11] to create novel content and context-aware music browsing interfaces. We have tried to combine our previous experience with knowledge from state-of-the-art systems in this domain to design a flexible framework to explore this new and fascinating interface design problem.

3. SYSTEM ARCHITECTURE

The goal of *Audioscapes* is to design a framework for exploring content and context-aware user interfaces for browsing large audio collections using controllers beyond the mouse and keyboard. The abstract system architecture distills the common functional blocks required to build such interfaces. A large number of existing audio and music browsing systems fit this architecture which is shown in Figure 1. The underlying metaphor is that each track in an audio collection is mapped to a discrete location on a rectangular grid. More than one track can be mapped to the same location. Different algorithms for clustering and dimensionality reduction can be used to map automatically extracted audio features to the grid coordinates. Controllers take input from the user and either interact with the audio collection (for example by initiating playback or by applying digital audio effects such as pitch-shifting and time-stretching) or move around on the mapped grid representing the audio collection. Views are different ways/devices used to display the grid map. The communication between processing blocks is mainly accomplished through Open Sound Control (OSC) [13] messages or alternatively custom XML or text files.

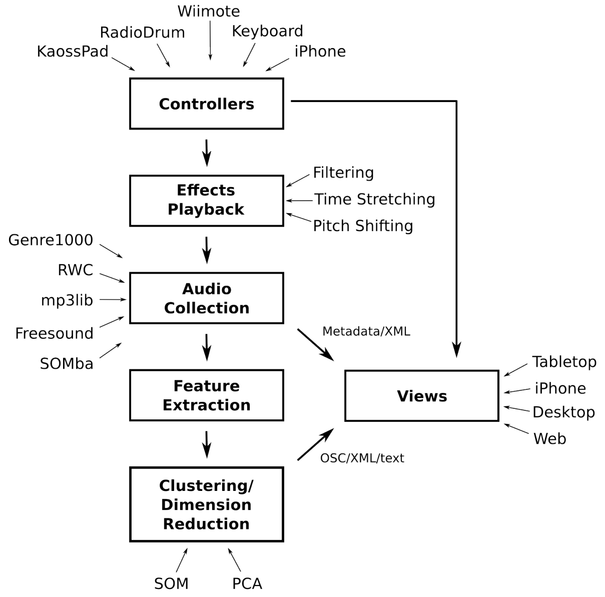


Figure 1. *System Architecture*

3.1. Audio Processing

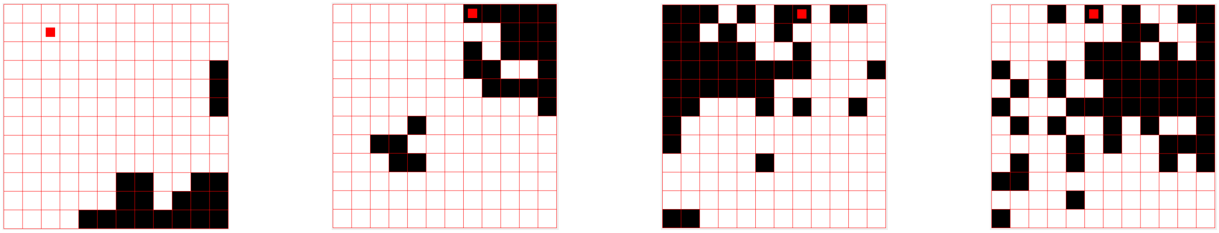
There are two main aspects of audio processing: digital audio effects and audio feature extraction. For digital audio effects we currently support pitch-shifting and time-stretching using a phasevocoder as well as tunable filters. The goal of audio feature extraction is to represent each song in a music collection as a single vector of features that characterize musical content. Using suitable features, songs that “sound” similar should have vectors that are “close” in the high dimensional feature space. The features used are the Spectral Centroid, Rolloff, Flux and the Mel-Frequency Cepstral Coefficients (MFCC) as well as time-domain zero crossings.

A more detailed description of the features and their motivation can be found in Tzanetakis and Cook [12]. This feature set has shown state-of-the-art performance in audio retrieval and classification tasks in the Music Information Retrieval Evaluation Exchange (MIREX) 2008 [1].

3.2. Visualization

The primary method used for visualization is the self organizing map (SOM) which is a type of neural network used to map a high dimensional input feature space to a lower dimensional representation while preserving the topology of the high dimensional feature space. This facilitates both similarity quantization and visualization simultaneously. The SOM was first documented in 1982 by T. Kohonen, and since then, it has been applied to a wide variety of diverse

[1] http://www.music-ir.org/mirex/



(a) Classical (b) Metal (c) Hiphop (d) Rock

Figure 2. Topological mapping of musical content by the Self-Organizing Map

clustering tasks [6]. In our system the SOM is used to map the audio features (64-dimensions) to two discrete coordinates on a rectangular grid. The traditional SOM consists of a 2D grid of neural nodes each containing an n-dimensional vector, $\mathbf{x}(\mathbf{t})$ of data. The goal of learning in the SOM is to cause different neighbouring parts of the network to respond similarly to certain input patterns. The network must be fed a large number of example vectors that represent, as closely as possible, the kinds of vectors expected during mapping. The data associated with each node is initialized to small random values before training. During training, a series of n-dimensional vectors of sample data are added to the map. The "winning" node of the map, known as the *best matching unit* (BMU), is found by computing the distance between the added training vector and each of the nodes in the SOM. This distance is calculated according to some pre-defined distance metric which in our case is the standard Euclidean distance on the normalized feature vectors.

Once the winning node has been defined, it and its surrounding nodes reorganize their vector data to more closely resemble the added training sample. The training utilizes competitive learning. The weights of the BMU and neurons close to it in the SOM lattice are adjusted towards the input vector. The magnitude of the change decreases with time and with distance from the BMU.

Figure 2 illustrates the ability of the extracted musical content-features and the SOM to represent musical content. The top subfigures (a), (b), (c) and (d) show how different musical genres are mapped to different regions of the SOM grid (the black squares are the ones containing one or more songs from each specific genre). As can be seen Classical, Heavy Metal and HipHop are well-localized and distinct whereas Rock is more spread out reflecting its wide diversity. The SOM is trained on a collection of 1000 songs spanning 10 genres. It is important to note that in all these cases the only information used is the automatically analyzed actual audio signal and the locations of the genres are emergent properties of the SOM.

3.3. View and Control Interfaces

The common functionality among view interfaces is to display the automatically calculated grid, respond to navigation events and handle audio playback and effects. Typically the grid squares are colored darker or lighter based on the number of tracks that they contain. The most powerful view is a desktop graphical user interface written in Qt [2]. In addition to standard view functionality it provides the ability to write iTunes music library XML files, advanced coloring modes based on metadata, and continuous playback mode in which tracks change automatically when the cursor moves to a different grid square without requiring explicit clicking by the user. In addition we also provide a web-interface that although more limited has the advantage that anyone on the internet can access and interact with the particular *AudioScape* deployed. As the audio is streamed, the audio collection remains on the server, this can be an important factor in commercial applications.

In order to explore non-standard form factors we have developed a implementation specific to the iPhone [3]. Having a touch-based display surface facilitates spatial awareness especially for blind or limited vision users. As the user moves her finger across the various squares, songs from each corresponding node cross-fade with each other to help her navigate the music collection by hearing how the songs in each grid location are changing. By laying out a music collection in this spatial fashion, navigation with only the knowledge of a few reference points is needed. In addition to the traditional mouse/keyboard based control, we have explored various alternative control interfaces. The Radiodrum [10] is a three dimensional controller that uses capacitive sensing to detect the positions of two radio frequency oscillators, usually attached to drum sticks or other similar objects. In our prototype, we have mapped the x, y and z axes of the sticks of the Radiodrum to the surface user interface. Movement of the sticks in the x and y axes moves the audio track selector cursor on the GUI, and movement in z controls the volume of that track. Each stick controls a different music track.

[2] http://www.qtsoftware.com/products
[3] http://www.apple.com/iphone

3.4. Data Collections and Implementation

In order to explore different configurations we have created *AudioScapes* for several large audio data collections which are known to the MIR and Computer Music community. The Freesound Project [4] is a huge collection of sound effects, music and environmental songs all licenced under the Creative Commons licence. The RWC (Real World Computing) Music Database [4] is a database of music that has been copyright cleared and made available to the Music Information Retrieval community. The music in the RWC database is from a wide variety of genres, with many classical and jazz pieces, as well as a sampling of the genres of popular, rock, dance, jazz, latin, classical, marches, world music, vocals, and traditional Japanese music. A collection of 1000 music tracks from 10 genres was gathered and described in [12]. We have also used a large database of 3000 30-second snippets from the personal collection of one of the authors. The Marsyas [5] audio processing software framework has been used for the audio feature extraction, digital audio effects, calculation of the SOM, the desktop graphical user interface and handling of controller data. For the web based interface we employ an *XHTML/CSS* and *Flash* based interface. The Open Sound Control (OSC) protocol [13] is used in this project to facilitate communication between the various components of the system.

4. DISCUSSION

AudioScapes is an extensible framework and architecture for surface-based interfaces for browsing large audio and music collections. Given the exploratory nature of the work we have not yet been able to conduct detailed quantitative user studies which are planned for the future. It is our hope that the developed interfaces have the potential to make browsing of audio collections much more effective, especially for users with special needs. We have been particularly fortunate to receive initial feedback from two blind users, one user with limited vision and one user with limited mobility. In all cases they were very positive about the system and provided valuable advice. As it is difficult to convey how the system works in paper we have collected videos and web demonstrations on a web-page `http://audioscapes.sness.net`.

5. ACKNOWLEDGEMENTS

National Science and Research Council of Canada (NSERC) funded parts of this work. Manjinder Benning, Darren Minifie, Allan Kumka, Jennifer Murdoch and Stephen Hitchner worked on different aspects of the framework.

[4] http://freesound.org

[5] http://marsyas.sourceforge.net

6. REFERENCES

[1] M. Cooper, J. Foote, E. Pampalk, and G. Tzanetakis, "Visualization in audio-based music information retrieval," *Computer Music Journal*, vol. 30, no. 2, pp. 42–62, 2006.

[2] F. Golshani, "TUI or GUI–it's a matter of somatics," *Multimedia, IEEE*, vol. 14, no. 1, 2007.

[3] M. Goto and T. Goto, "Musicream: New music playback interface for streaming, sticking, sorting, and recalling musical pieces," in *Proc. Int. Conf. on Music Information Retrieval (ISMIR)*, 2005.

[4] M. Goto and T. Nishimura, "RWC music database: Music genre database and musical instrument sound database," in *Proc. Int. Conf. on Music Information Retrieval (ISMIR)*, 2003.

[5] H. Ishii and B. Ullmer, "Tangible bits: towards seamless interfaces between people, bits and atoms," in *Proc. SIGCHI*, 1997.

[6] T. Kohonen, *Self-Organizing Maps*, ser. Springer Series in Information Sciences, Berlin, Heidelberg, 1995, vol. 30.

[7] F. Mörchen, A. Ultsch, M. Nöcker, and C. Stamm, "Databionic visualization of music collections according to perceptual distance," in *Proc. Int. Conf. on Music Information Retrieval (ISMIR)*, 2005.

[8] J. Murdoch and G. Tzanetakis, "Interactive content-aware music browsing using the radio drum," in *Proc. IEEE Int. Conf. on Multimedia and Expo (ICME)*, 2006.

[9] E. Pampalk, S. Dixon, and G. Widmer, "Exploring music collections by browsing different views," in *Proc. Int. Conf. on Music Information Retrieval (ISMIR)*, 2003.

[10] W. Schloss and P. Driessen, "New algorithms and technology for analyzing gestural data," in *Proc. IEEE Pacific Rim Conference*, 2001.

[11] J. M. Stephen Hitchner and G. Tzanetakis, "Music browsing using a tabletop display," in *Proc. Int. Conf. on Music Information Retrieval (ISMIR)*, 2007.

[12] G. Tzanetakis and P. Cook, "Musical Genre Classification of Audio Signals," *IEEE Trans. on Speech and Audio Processing*, vol. 10, no. 5, Jul. 2002.

[13] M. Wright, A. Freed, and A. Momeni, "Opensound control: State of the art 2003," in *Int. Conf. on New Interfaces for Musical Expression (NIME'03)*, Montreal, Canada, 2003.