Beyond the Query-By-Example Paradigm: New Query Interfaces for Music Information Retrieval

George Tzanetakis, Andreye Ermolinskyi, Perry Cook
Computer Science Department, Princeton University
email: gtzan@cs.princeton.edu

Abstract

The majority of existing work in music information retrieval for audio signals has followed the content-based query-by-example paradigm. In this paradigm a musical piece is used as a query and the result is a list of other musical pieces ranked by their content similarity. In this paper we describe algorithms and graphical user interfaces that enable novel alternative ways for querying and browsing large audio collections. Computer audition algorithms are used to extract content information from audio signals. This automatically extracted information is used to configure the graphical user interfaces and to genereate new query audio signals for browsing and retrieval.

1 Introduction

Audio and especially music signals constitute a significant part of internet traffic. Due to advances in storage capacity, network bandwidth and audio compression technology computer users today can archive large collections of audio signals. Existing audio software tools are inadequate to handle these collections of increasing size and complexity as they are developed for music recording and production purposes and are centered around the notion of processing a single file. In order to enable novel ways of retrieving, structuring, and interacting with large audio collections new algorithms and tools need to be designed and developed.

The advantages of efficient searching and retrieval of text are evident from the popular web search engines based on Information Retrieval and Machine Learning techniques. More recently similar methods have been proposed for images spurred by the wide spread use of digital photography. Obviously text retrieval techniques can also be used for multimedia data such as audio and images using manual metadata information such as filenames. However this approach by itself is inadequate for effective search and retrieval. Audio signals and especially musical signals are complex data-intensive dynamic signals that have unique characteristics and requirements. Unfortunately no widely used algorithms and systems for searching and retrieving audio signals have been developed.

In recent years, techniques for audio and music information retrieval have started emerging as research prototypes. These systems can be classified into two major paradigms. In the first paradigm the user sings a melody and similar audio files containing that melody are retrieved. This approach is called "Query by Humming" (QBH). Unfortunately it has the disadvantage of being applicable only when the audio data is stored in symbolic form such as MIDI files. The conversion of generic audio signals to symbolic form, called polyphonic transcription, is still an open research problem in its infancy. Another problem with QBH is that it is not applicable to several musical genres such as Dance music where there is no singable melody that can be used as a query. In the second paradigm called "Query-by-Example" (QBE) an audio file is used as the query and audio files that have similar content are returned ranked by their similarity. In order to search and retrieve general audio signals such as mp3 files on the web only the QBE paradigm is currently applicable.

In this paper we propose new ways of browsing and retrieving audio and musical signals from large collections that go beyond the QBE paradigm. The developed algorithms and tools rely on the automatic extraction of content information from audio signals. The main idea behind this work is to create new audio signals either by combination of other audio signals, or synthetically based on various constraints. These generated audio signals are subsequently used to auralize queries and parts of the browsing space. The ultimate goal of this work is to lay the foundations for the creation of a musical "sketchpad" which will allow computer users to "sketch" the music they want to hear. Ideally we would like the audio equivalent of sketching a green circle over a brown rectangle and retrieving images of trees (something which is supported in current content-based image retrieval systems).

In addition to going beyond the QBE paradigm for audio and music information retrieval this work differs from the majority of existing work in two ways: 1) continuous aural feedback 2) use of computer audition. Continuous aural feedback means that the user constantly hears audio or music that corresponds to her actions. Computer audition techniques extract content information from audio signals that is used to configure the graphical user interfaces.

The term Query User Interfaces (QUI) will be used in this paper to describe any interface that can be used to specify in some way audio and musical aspects of the desired query. Two major families of Query Interfaces will be described based on the feedback they provide to the user. The first family consists of interfaces that utilize directly audio files in order to provide feedback while the second family consists of interfaces that generate symbolic information in MIDI format. It is important to note that in both of these cases the goal is to retrieve from general audio and music collections and not symbolic representations. For the remainder of the paper it will be usefull to imagine a hypothetical scenario where all of recorded music is available digitally and the goal is to structure, search and retrieve from this large musical universe. It is likely that this hypothetical scenario will become reality in the near future.

This paper is structured as follows: Section 2 describes previous and related work. Computer audition techniques that automatically extract information from audio signals are briefly discussed in Section 3. Content and context aware displays for audio signals and collections are covered in Section 4. Section 5 is about audio-based QUIs and Section 6 is about MIDI-based QUIs.

2 Related work

In recent years several academic and commercial systems have been proposed for content-based retrieval of images. Some representative examples are the Blobword system (Belongie et al. 1998) developed at UC Berkeley, the PhotoBook from MIT (Pentland et al. 1994) and the QBIC system from IBM (Flickner and et al. 1995). In these systems the user can search, browse and retrieve images based on similarity and various automatic feature extraction methods.

One of the earliest attempts for a similar system for audio is described in (Wold et al. 1996). The Sonic Browser from the University of Limerick, Ireland (Fernstrom and Brazil 2001) is a graphical user interface based on direct manipulation and sonification for browsing collections of audio signals which are central concepts behind the design of the interfaces described in this paper.

Direct manipulation systems visualize objects and actions of interest, support rapid, reversible, incremental actions, and replace complex command-language syntax by direct manipulation of the object of interest (Shneiderman 1998). Direct sonification refers to the immediate aural feedback to user actions. Examples of such direct manipulation systems include the popular desktop metaphor, computer-assisted-design tools, and video games. The main property of direct manipulation interfaces is the immediate aural and visual feedback in response to the user actions. The design of the interfaces described in this paper was guided by the Schneiderman mantra for direct manipulation information visualization systems: "overview first, zoom and filter, then details on demand". In this work our main focus is the filtering stage.

Another important influence for this work is the legacy of systems for automatic music generation and style modelling. These systems typically fall into four major categories: generating music (Biles 1994), assisting composition (Masako Nishijima and Watanabe 1992), modelling style, performance and/or composers (Garton 1992), and automatic musical accompaniment (Dannenberg 1984). These references are indicative and representative of early work in each catagory. There are also commercial systems that generate music according to a variety of musical styles such as the *Band in a box* software: `http:://www.sonicspot.com/bandinabox/bandinabox.html`

3 Computer Audition Techniques

Currently the majority of existing tools for interacting with audio collections rely on metadata information such as the filename or ID3 tags in order to organize, structure and retrieve audio data. As audio collections become increasing larger and more complex, more sophisticated modes of interaction are desired. Metadata information is inadequate by itself to support more sophisticated interactions for two reasons. The first is that manual acquisition of metadata is time consuming especially if more complex information than just the artist and the musical genre is required. The second limitation is that certain audio and musical properties are impossible to accurately capture using manual metadata. For example the tempo of a song can only be grossly characterized (fast, medium, slow for example) by human users unless they are musically trained. However, automatic tempo estimation can be performed quite accurately for certain types of music.

The term Computer Audition (CA) will be used in this paper to describe any algorithm and tool that automatically extracts information from audio signals. In recent years a large variety of CA algorithms have been proposed. These techniques build upon ideas from the fields of Signal Processing, Machine Learning, and Information Retrieval in order to extract information from audio and especially music signals. In this section we review some of the current state of the art algorithms in Computer Audition. The main focus is mostly the information that is extracted and will be subsequently used to create novel user interfaces rather than how the algorithms work. More details about the algorithms can be found in the corresponding references.

The basis of most proposed CA algorithms is the extraction of numerical features to characterize either short segments of audio or full audio files. These features are typically calculated based on Signal Processing and Pattern Recognition techniques that analyze the statistics of the energy distribution of the signal in time and frequency. For the purposes of this section we will assume that the audio features have been extracted and are used as the representation for audio signals that is subsequently analyzed.

The most fundamental computer audition technique that has to be supported is query-by-example content-based similarity retrieval. One of the first papers describing this process is (Wold et al. 1996) where isolated sound such as instrument tones and sound effects are retrieved by similarity. Essentially similarity retrieval is based on computing distances between feature vectors.

Another important computer audition technique is classification. In classification an audio signal is automatically assigned to a label from a predefined set of class labels. Various types of audio classification have been proposed in the literature. Some representative examples are: music vs speech (Scheirer and Slaney 1997), isolated musical instrument sounds and sound effects (Wold et al. 1996), and musical genres (Tzanetakis and Cook 2002). Automatic beat extraction has been studied in (Scheirer 1998; Goto and Muraoka 1998; Laroche 2001) and used for automatic musical genre classification in (Tzanetakis and Cook 2002).

Audio thumbnailing refers to the process of producing short, representative samples (or "audio thumbnails") of a larger piece of music. Thumbnailing based on analysis of Mel Frequency Cepstral Coefficients (clustering, hidden-markov models (HMM)) was explored in (Logan 2000). A chroma transformation of the spectrum is used in (Bartsch and Wakefield 2001) to find repeating patterns that correspond to the chorus of popular music songs which is used as an audio thumbnail.

For the remainder of the paper it will be assumed that there is a collection of audio files that have been analyzed and each file is annotated with various continuous and discrete attributes such as tempo, spectral features, musical genre labels etc. In addition structural information such as thumbnail, segmentation and looping points are also atuomatically extracted. Of course it is also possible to acquire some of this information manually (for example some of the discrete attributes). Finally it is important to emphasize that although in most cases the attributes can be precomputed, CA algorithms are still required to process the generated audio queries and therefore have to be integrated in the system.

4 Content and Context Aware Displays

Content and context aware displays for audio signals represent audio signals and collections as visual objects with properties that reflect information about the audio signal. Content information refers to information that depends only on a specific audio signal such as tempo while context information refers to information about the relation of the audio signal to a collection. As an example of context information in a large collection of all musical styles , two female singer Jazz pieces would be similar whereas in a collection of vocal Jazz they might be quite dissimilar.

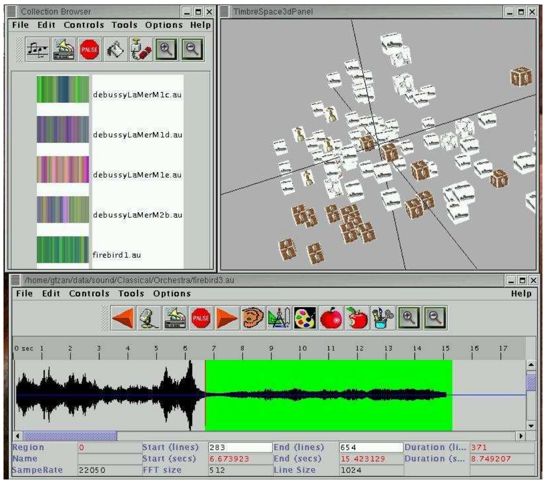

Figure 1: Content-context displays.

Some examples of such content and context aware displays are *Timbrespaces* which are 2D or 3D spaces containings visual objects such that each object corresponds to a sound file. They can be constructed so that visual proximity of objects corresponds to audio content similarity and appearance corresponds to audio content information. *Timbregrams* are a static visualization of sound files that shows time periodicity and similarity structure of audio signals using colors. Both of these displays are content and context aware and are based on automatic feature extraction and dimensionality reduction techniques such as Principal Component Analysis (PCA). More information about them can be found in (Tzanetakis and Cook 2001). Figure 1 shows Timbregrams on the left and a Timbrespace on the right as well as a traditional audio waveform display at the bottom. Although these displays are not the main focus of this paper they are mentioned as the query interfaces which will be described are used in conjuction with them for audio browsing and retrieval.

5 Audio-based Query Interfaces

In audio and music information retrieval there is a query and a collection that is searched for similar files. The main idea behind audio-based QUIs is to utilize not only the query's audio but also utilize directly the audio collection. Since the structure and content of the collection are already automatically or manually extracted for retrieval, this information can be used together with the audio to provide interactive aural feedback to the user. This idea will be clarified in the following sections with examples of audio-based QUIs.

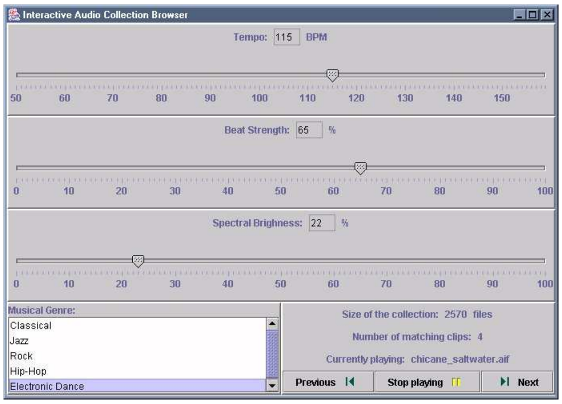

Figure 2: Sound sliders and palettes.

5.1 Sound Sliders

Sound sliders are used to browse audio collections based continuous attributes. For presentation purposes assume that for each audio file in a collection the tempo and beat strength of the song are automatically extracted. For each of these two attributes a corresponding *sound slider* is created. For a particular setting of the sliders the sound file with attributes corresponding to the slider values is selected and part of it is played in a loop. If more than one sound files correspond to the particular slider values, the user can advance in circular fashion to the next sound file by pressing a button. One way to view this is that for each particular setting of slider values there is a corresponding list of sound files that have corresponding attributes. When the sliders are moved the sound is crossfaded to a new list that corresponds to the new slider settings. The extraction of the continuous attributes and their sorting is performed automatically.

In current audio software typically sliders are first adjusted, and then by pressing a submit button, files that correspond to these parameters are retrieved. Unlike this traditional use of sliders for setting parameters, sound sliders provide continuous aural feedback that corresponds to the actions of the user (direct sonification). So for example when the user sets the tempo to 150 beats-per-minute (bpm) and beat strength to its highest value there is immediate feedback about what the values represent by hearing a corresponding fast song with strong rhythm. Another important aspect about *sound sliders* is that they are not independant and the aural feedback is influenced by the settings of all of them. Figure 2 shows a screenshot of sound sliders used in our system.

5.2 Sound palettes

Sound palettes are similar to sound sliders but apply to browsing discrete attributes. A palette contains a fixed set of visual objects (text, images, shapes) that are arranged based on discrete attributes. At any time only one of the visual objects can be selected by clicking on it with the mouse. For example objects might be arranged in a table by genre and year of release. Continuous aural feedback similar to the sound sliders is supported. The bottom left corner of Figure 2 shows a content palette corresponding to musical genres. Sound palettes and sliders can be combined in arbitrary ways.

5.3 Loops

In the previously described query interfaces it is necessary to playback sound files continuously in a looping fashion. Several methods for looping are supported in the system. The simplest method is to loop the whole file with crossfading at the loop point. The main problem problem with this method is that the file might be too long for browsing purposes. For files with a regular rhythm, automatic beat detection tools are used to extract loops typically corresponding to an integer number of beats. Another approach that can be applied to arbitrary files is to loop based on spectral similarity. In this approach the file is broken to short windows and for each window a numerical representation of the main spectrum characteristics is calculated. The file is searched for windows that have similar spectral characteristics and these windows can be used to achieve smooth looping points. Finally automatic thumbnail methods that utilize more high level information can be used to extract representative short thumbnails.

5.4 Music Mosaics

Loops and thumbnail techniques can be used to create short representations of audio signals and collections. Another possibility for the representation of audio collections is the creation of *Music Mosaics* that are pieces created by concatenating short segments of other pieces (Schwarz 2000; Zils and Pachet 2001). For example in order to represent a collection of songs by the Beatles, a music mosaic that would sound like Hey Jude could be created by concatenating small pieces from other Beatles songs. Time-stretching based on beat detection and overlap-add techniques can also be used in *Music Mosaics*.

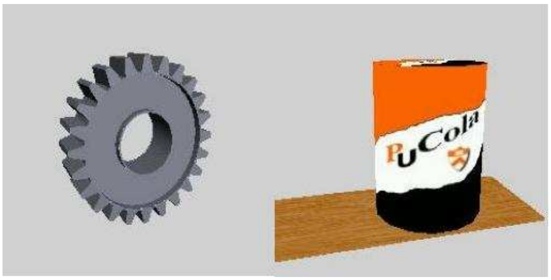

Figure 3: 3D sound effects generators.

5.5 3D sound effects generators

In addition to digital music distribution, another area where large audio collection are utilized is libraries of digital sound effects. Most of the times we hear a door opening or a telephone ringing in a movie the sounds are not actually recorded during the filming of the movie but are taken from libraries of prerecorded sound effects.

Searching libraries of digital sound effects poses new challenges to audio information retrieval. Of special interest are sound effects where the sound depends on the interaction of a human with a physical object. Some examples are: the sound of walking on gravel or the rolling of a can on a wooden table. In contrast to non-interactive sound effects such as a door bell the production of those sounds is closely tied to a mechanical motion.

In order to go beyond the QBE paradigm for content retrieval of sound effects, query generators that somehow model the human production of these sounds are desired. Towards this goal we have deloped a number of interfaces with the common theme of providing a user-controlled animated object connected directly to the parameters of a synthesis algorithm for a particular class of sound effects. These 3D interfaces not only look similar to their real world counterparts but also sound similar. This work is part of the Physically Oriented Library of Interactive Sound Effects (PhOLISE) project (Cook 1997). This project uses physical and physically motivated analysis and synthesis algorithms such as modal synthesis, banded waveguides (Essl and Cook 2000), and stochastic particle models (Cook 1999) to provide interactive parametric models of real-world sound effects.

Figure 3 shows two such 3D sound effects query generators. The PuCola Can shown on the right is a 3D model of a soda can that can be slid across various surface textures. The sliding speed and texture material are controlled by the user and result in appropriate changes to the sound in real-time. The Gear on the left is a real-time parametric synthesis of the sound of a turning wrench. Other developed generators are: a model of falling particles on a surface where parameters such as density, speed, and material are controlled, and a model of walking sounds where parameters such as gait, heel and toe material, weight and surface material are controlled.

6 MIDI-based Query Interfaces

In constrast to audio based QUIs, MIDI-based QUIs do not utilize directly the audio collection but rather synthetically generate query audio signals from a symbolic representation (MIDI) that is created based on the user actions. The parameters used for the generation of this query and/or the automatically analyzed generated audio are used to retrieve similar audio pieces from a large collection. It is important to note that although MIDI is used for the generation of the queries, these interfaces are used for searching audio collections. These concepts will be clarified in the subsequent sections with concrete examples of MIDI-based QUIs.

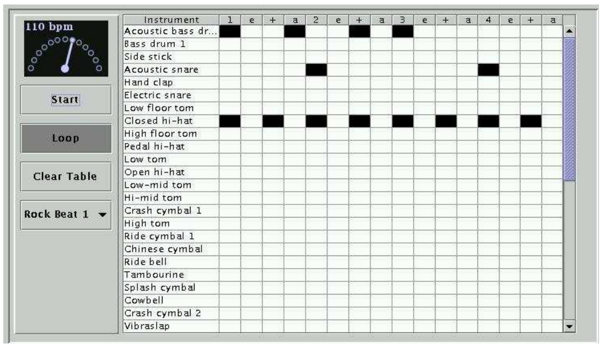

Figure 4: Groove box

6.1 The groove box

The groove box is similar to standard software drum machine emulators where beat patterns beat patterns are created using a rectangular grid of note values and drum sounds. Figure 4 shows a screenshot of a groove machine where the user can create a beat pattern or select it from a set of predefined patterns and speed it up and down. Soundfiles are retrieved based on the interface settings as well as by audio analysis (Beat Histograms (Tzanetakis and Cook 2002)) of the generated beat pattern. Another possibity is to use beat analysis methods based on extracting specific sound events (such as bass drum and cymbal hits) and match each drum track separately. We are also investigating the possibility of automatically alligning the generated beat pattern with the retrieved audio track and playing both as the user edits the beat pattern. The code for the groove box is based on demonstration code for the JavaSound API.

6.2 The tonal knob

The tonal knob shows a circle of fifths which is a ordering of the musical pitches so that harmonically related pitches are successively spaced in a circle. The user can select the tonal center and hear a chord progression at a specified musical style that establishes that tonic center. Pieces with the same tonal center are subsequently retrieved using Pitch Histograms ((Tzanetakis and Cook 2002)).

6.3 Style machines

Style machines are more close to standard automatic music generation interfaces. For each particular style we are interested in modelling for retrieval a set of sliders and buttons corresponding to various attributes

Figure 5: Indian music style machine

of the style are used to generate in real-time an audio signal. This signal is subsequently used as a query for retrieval purposes. Style attributes include tempo, density, key center, and instrumentation. Figure 5 shows a screenshot of *Ragamatic*, a style machine for Indian Music.

7 Implementation

All the graphical user interface components described in this paper are implemented in Java. The Java3D API is used for the 3D graphics and JavaSound API is used for the audio and MIDI playback. The computer audition algorithms are performed using MARSYAS (Tzanetakis and Cook 2000) a free object-oriented framework for audio analysis available under the GNU public licence from `http://www.cs.princeton.edu/~gtzan/marsyas.html`. The parametric synthesis of soundeffects and audio generation from MIDI is performed using the Synthesis Toolkit (Cook and Scavone 1999) `http://ccrma-www.stanford.edu/software/stk/`.

The various components of the system are connected following a client-server application that allows the possibility of a distributed system. The use of portable ANSI C++ and JAVA enables the system to compile in a variety of operating systems and configurations. The software has been tested on Linux, Solaris, Irix and Windows (Visual Studio and Cygwin) systems.

8 Discussion

Although the described graphical user interface components were presented separately they are all integrated following a Model-View-Controller user interface design framework (Krasner and Pope 1988). The Model part comprises of the actual underlying data and the operations that can be performed to manipulate it. The View part describes specific ways the data model can be displayed and the Controller part describes how user input can be used to control the other two parts. By sharing the Model part the graphical user interface components can affect the same underlying data. That way for example *sound sliders*, *sound palettes*, and *style machines* can all be used together to browse the same audio collection visualized as a *Timbrespace*.

Of course in a complete music information retrieval system traditional graphical user interfaces for searching and retrieval such as keyword or metadata searching would also be used in addition to the described interfaces. These techniques and ideas are well-known and therefore were not described in this paper but are part of the system we are developing.

Although the description of the system in this paper has emphasized music retrieval it is clear that such a system would also be useful for music creation. For example sound designers, composers (especially utilizing ideas from Music Concrete and Granular Synthesis), and DJs would all benefit from novel ways of interacting with large collections of audio signals.

It is our belief that the problem of query specification has many interesting directions for future research because it provides a new perspective and design constraints to the issue of automatic music and sound generation and representation. Unlike existing work in automatic musical generation where the goal is to create as good music as possible, the goal in our work is to create convincing sketches or caricatures of the music that can provide feedback and be used for music information retrieval. Another related issue is the visual design of interfaces and displays and its connection to visual music and score representations.

9 Future work

Evaluating a complex retrieval system with a large number of graphical user interfaces is difficult and can only be done by conducting user studies. For the future we plan a task-based evaluation of our system similar to (Jose, Furner, and Harper 1998) where users will be given a specific task such as locating a particular song in a large collection and their performance using different combinations of tools will be measured.

Another direction for future research is the collaboration with composers and music cognition experts with the goal of exploring different metaphors for sketching music queries for users with and without musical background and how those differ. Currently a lot of information about audio and especially music signals can be found in the form of metadata annotations such as filenames and ID3 tags of mp3 files. We plan to develop web crawlers that will collect that information and use it to build more effective tools for interacting with large audio collections.

10 Summary

A series of graphical user interface component that enhance and extend the standard query-by-example paradigm for music information retrieval were presented. They are based on ideas from direct manipulation graphical user interfaces and automatic computer audition techniques for extracting information from audio signals. They can be seen as a first step towards providing the equivalent of a digital sketchpad interface for browsing audio collections and generating audio queries in the context of music information retrieval.

11 Acknowledgments

Doug Turnbull did initial work on Music Mosaics. This work was funded under NSF grant 9984087, State of New Jersey Commission on Science and Technology grant 01-2042-007-22, and from gifts from Intel and Arial Foundation.

References

Bartsch, M. A. and G. H. Wakefield (2001). To Catch a Chorus: Using Chroma-Based Representation for Audio Thumbnailing. In *Proc. Int. Workshop on applications of Signal Processing to Audio and Acoustics*, Mohonk, NY, pp. 15–19. IEEE.

Belongie, S., C. Carson, H. Greenspan, and J. Malik (1998, January). Blobworld: A system for region-based image indexing and retrieval. In *Proc. 6th Int. Conf. on Computer Vision*.

Biles, J. (1994, September). GenJam: A Genetic Algorithms for Generating Jazz Solos. In *Proc. Int. Computer Music Conf. (ICMC)*, Aarhus, Denmark, pp. 131–137.

Cook, P. (1997, August). Physically inspired sonic modeling (PHISM): synthesis of percussive sounds. *Computer Music Journal 21*(3).

Cook, P. (1999). Toward physically-informed parametric synthesis of sound effects. In *Proc. IEEE Workshop on applications of Signal Processing to Audio and Acoustics, WASPAA*, New Paltz, NY. Invited Keynote Address.

Cook, P. and G. Scavone (1999, October). The Synthesis Toolkit (STK), version 2.1. In *Proc. Int. Computer Music Conf. ICMC*, Beijing, China. ICMA.

Dannenberg, R. (1984). An on-line algorithm for real-time accompaniment. In *Proc. Int. Computer Music Conf.*, Paris, France, pp. 187–191.

Essl, G. and P. Cook (2000). Measurements and efficient simulations of bowed bars. *Journal of Acoustical Society of America (JASA) 108*(1), 379–388.

Fernstrom, M. and E. Brazil (2001, July). Sonic Browsing: an auditory tool for multimedia asset management. In *Proc. Int. Conf. on Auditory Display (ICAD)*, Espoo, Finland.

Flickner, M. and et al. (1995, September). Query by image and video content: the QBIC system. *IEEE Computer 28*(9), 23–32.

Garton, B. (1992, October). Virtual Performance Modelling. In *Proc. Int. Computer Music Conf. (ICMC)*, San Jose, California, pp. 219–222.

Goto, M. and Y. Muraoka (1998). Music Understanding at the Beat Level: Real-time Beat Tracking of Audio Signals. In D. Rosenthal and H. Okuno (Eds.), *Computational Auditory Scene Analysis*, pp. 157–176. Lawrence Erlbaum Associates.

Jose, J. M., J. Furner, and D. J. Harper (1998). Spatial querying for image retrieval: a user-oriented evaluation. In *Proc. SIGIR Conf. on research and development in Information Retrieval*, Melbourne, Australia. ACM.

Krasner, G. E. and S. T. Pope (1988, August). A cookbook for using the model-view-controller user interface paradigm in Smalltalk-80. *Journal of Object-Oriented Programming 1*(3), 26–49.

Laroche, J. (2001). Estimating Tempo, Swing and Beat Locations in Audio Recordings. In *Proc. Int. Workshop on applications of Signal Processing to Audio and Acoustics WASPAA*, Mohonk, NY, pp. 135–139. IEEE.

Logan, B. (2000). Music summarization using key phrases. In *Proc. Int. Conf. on Acoustics, Speech and Signal Processing ICASSP*. IEEE.

Masako Nishijima and K. Watanabe (1992). Interactive Music Composer based on Neural Networks. In *Proc. Int. Computer Music Conf. (ICMC)*, San Jose, California.

Pentland, A., R. Picard, and S. Sclaroff (1994, July). Photobook: Tools for Content-Based Manipulation of Image Databases. *IEEE Multimedia*, 73–75.

Scheirer, E. (1998, January). Tempo and beat analysis of acoustic musical signals. *Journal of the .Acoustical Society of America 103*(1), 588,601.

Scheirer, E. and M. Slaney (1997). Construction and evaluation of a robust multifeature speech/music discriminator. In *Proc. Int. Conf. on Acoustics, Speech and Signal Processing ICASSP*, pp. 1331–1334. IEEE.

Schwarz, D. (2000, December). A system for data-driven concatenative sound synthesis. In *Proc. Cost-G6 Conf. on Digital Audio Effects (DAFX)*, Verona, Italy.

Shneiderman, B. (1998). *Designing the User Interface: Strategies for Effective Human-Computer Interaction* (3rd ed. ed.). Addison-Wesley.

Tzanetakis, G. and P. Cook (2000). Marsyas: A framework for audio analysis. *Organised Sound 4(3)*.

Tzanetakis, G. and P. Cook (2001, August). Marsyas3D: a prototype audio browser-editor using a large scale immersive visual and audio display. In *Proc. Int. Conf. on Auditoy Display (ICAD)*, Espoo, Finland.

Tzanetakis, G. and P. Cook (2002). Musical Genre Classification of Audio Signals. *IEEE Transactions on Speech and Audio Processing*. (accepted for publication).

Wold, E., T. Blum, D. Keislar, and J. Wheaton (1996). Content-based classification, search and retrieval of audio. *IEEE Multimedia 3*(2).

Zils, A. and F. Pachet (2001, December). Musical Mosaicing. In *Proc. Cost-G6 Conf. on Digital Audio Effects (DAFX)*, Limerick, Ireland.