

Bringing Mobile Map Based Access to Digital Audio to the End User

SUBMITTED TO VMDL07 AND ICIAP07

Robert Neumayer, Jakob Frank, Peter Hlavac, Thomas Lidy and Andreas Rauber
Vienna University of Technology
Department of Software Technology and Interactive Systems
Favoritenstraße 9-11, Vienna, Austria
{neumayer,frank,lidy,rauber}@ifs.tuwien.ac.at

Abstract

Private as well as commercial music collections keep growing in size and diversity. With an increasing number of tracks and the resulting complexity users quickly face problems in handling their collections in an adequate way. At the same time, new business models of online vendors arise and the inherent industry interest in new ways of distribution channels and devices becomes immanent. In this paper, we show alternative ways of interacting with large music collections, based on the Self-Organising Map clustering algorithm applied to an audio feature representation of audio files. We therein focus on the presentation of full desktop applications as well as applications for mobile devices like PDAs and Smartphones with the goal of bringing Music Information Retrieval technologies closer to end users. Further, the presented interfaces give an outlook to means of access to other types of media in streaming environments, e.g. video.

1 Motivation

A wide range of digital objects have increasingly become to play a central part in many private users lives. In the music domain, recent years have witnessed drastic increases in size and numbers of both private and commercial audio collections being available in digital form.

The urgent need for adequate means of access becomes apparent by only looking at the sheer amount of music a growing number of people own or use today. The hierarchical browsing of meta data categories may have been a feasible option a few years ago when users had access to up to a few hundred songs, but is clearly not practicable anymore when thinking about the 60 GB of storage that e.g. *iPods* offer. Especially small devices and their limited means of

interaction, often reduced to simple click wheels or touch screens, are examples where usability could greatly benefit from alternative ways of interaction.

Keeping the oversight of a music collection comprising several thousand songs quickly becomes a challenge for every user, a problem becoming even more compelling in the case of online music vendors offering even larger collections. It is therefore a necessity to provide adequate means of access and to abstract from often poorly managed metadata. By using feature descriptions for audio, similarity search, i.e. the automatic recommendation of similar songs, offers possibilities to propose customised playlists to users. Rather than listening to all songs of one genre, users can choose to listen to songs similar to ones they like.

On the other hand, online music vendors face an even greater challenge when administering their collections. Moreover, they not only want to keep track of what content their repositories hold, but also have a strong interest in appropriate means of offering their tracks in terms of marketing their products. The usage of recommendation engines has proven to be a viable option in many other areas such as the general online shopping scenario as shown by Amazon.com. In order to satisfy their customers needs and to adopt new business models like music on demand, the music industry will have a strong need for intuitive interfaces, that possess the ability to abstract from the original piece of music and subsequently allowing for aggregation of information about a collection.

The area of Music Information Retrieval (MIR), i.e. the automatic description and use of audio for retrieval and classification tasks, has been heavily researched and significant advances have been made particularly over the last decade. Yet, the MIR community so far has not managed to provide adequately matured prototypes implementing the promising algorithms and techniques. Many applications have been developed, albeit only a few made it beyond the stage of re-

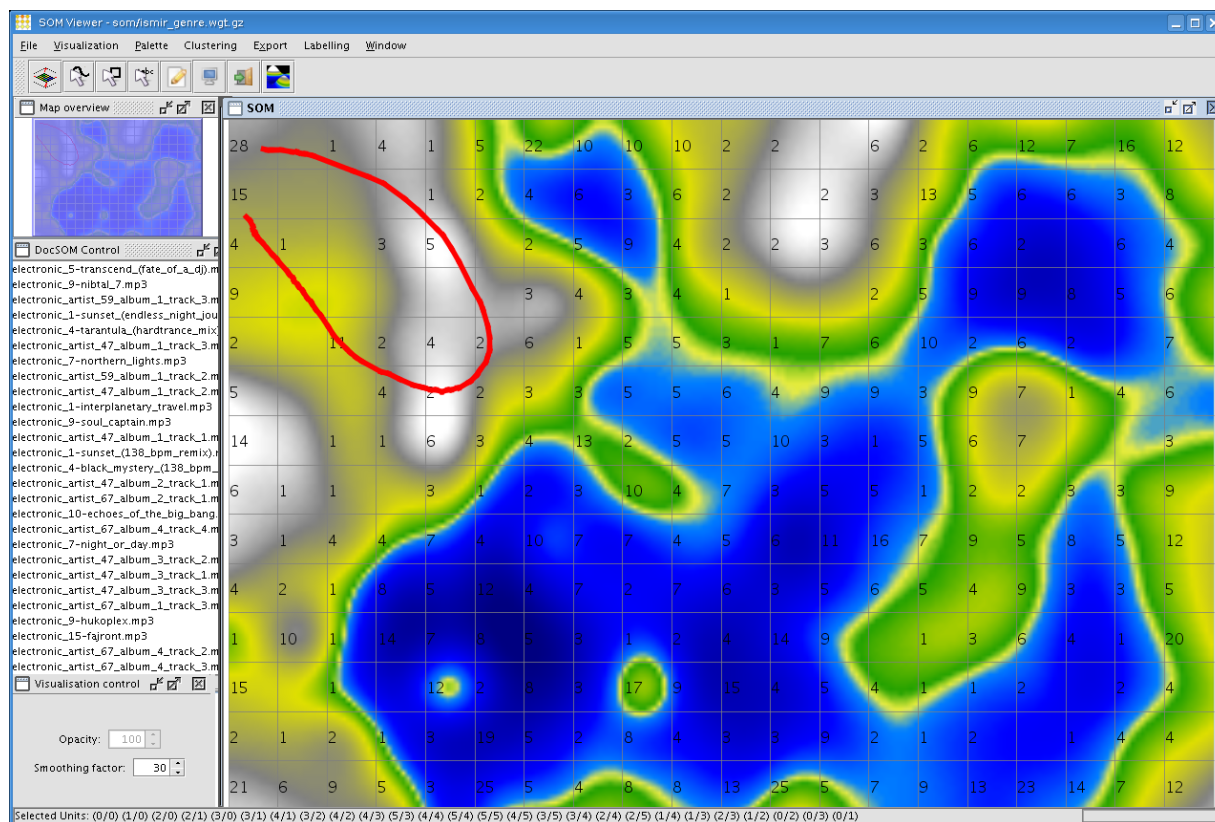


Figure 1. The main PlaySOM interface.

search prototypes. Even these are often not close enough to industrial strength prototypes to actually be of help to end users or integration in the current network infrastructure. Driven by the demand from both industry and private end users, intuitive user interfaces pose a key factor in both the mediation of research results and the propagation of techniques to end users and industry partners.

In this paper we will give an overview of techniques feasible for alternative ways of accessing music collection. As well as we are presenting desktop applications, we will focus on mobile applications making use of modern networking methods. We give an overview about prototypes for interfacing audio collections having reached a level of maturity so that they are usable and integrate with the current network infrastructure.

The remainder of this paper is structured as follows. Section 2 gives an overview of previous work done in the areas of Music Information Retrieval and covers the underlying methods used. We then introduce a range of user interfaces for both desktop computers and mobile devices in Section 3. Finally, we draw conclusions and give a brief outlook on future work in Section 4.

2 Related Work

2.1 Music Information Retrieval (MIR)

The basic idea of Music Information Retrieval as a research discipline is to provide techniques to appropriately index and search in collections of digital music. Therefore, different sets of audio features have been developed.

A range of feature sets for audio indexing exist, the MARSYAS system, for example, is an early implementation and implements a wide range of feasible audio features [9].

The algorithm for feature computation we rely on in this paper is denoted as *Rhythm Patterns*, and is described in detail in [7]. It is realised via the computation of a psycho-acoustically transformed spectrogram, i.e. a Bark-scale Sonogram reflecting human loudness sensation. In the second part of the algorithm, the spectrum is transformed via a discrete Fourier transform into a time-invariant representation of amplitude of loudness modulations on critical bands for different modulation frequencies. Patterns in this matrix representation reflect rhythmicity in the audio signal, thus the feature set is called *Rhythm Patterns*.

An alteration of this algorithm, explained in [4], resulted in a method for computation of *Statistical Spectrum Descriptors*, which yield relatively good results at a manageable dimensionality of 168 features as compared to the original *Rhythm Patterns* that comprise 1440 feature values. Other feature sets, such as Rhythm Histograms [4], can also be used as an input for the applications we describe in this paper.

The resulting audio feature vectors can be processed by machine learning algorithms, e.g. for music classification or clustering, and subsequently in more sophisticated applications based thereon.

2.2 Map Based User Interfaces

To create *maps of music*, different clustering algorithms can be applied to the vector representation of the audio files. The Self-Organising Map (SOM), an unsupervised neural network model, was first introduced by Kohonen and therefore is also referred to as Kohonen map [3]. The basic idea of clustering is the identification of coherent sub-groups of similar instances, i.e. pieces of audio, or rather data vectors representing them. The SOM combines this with a mapping of instances from this high-dimensional input space to a low-dimensional, usually two-dimensional, output space. Hence, a Self-Organising Map is a two-dimensional grid of $m \times n$ so called units, each represented by a randomly initialised vector of the same length as the input space. During its training process, each input vector is presented to the SOM and assigned to the closest unit in terms of a distance function, usually the Euclidean distance. Additionally, the closest matching unit vector (and, up to a certain extent, its neighbouring units) is moved towards the input vector as to even more closely resemble it in the next iteration. This process takes into account topology issues, i.e. songs close to each other in the input space, are mapped onto similar regions of the map. The result is a 2D representation of audio files, i.e. a map of music having a certain number of similar songs assigned to each of its units.

Since the SOM clustering itself does not convey any additional information that could help in a better understanding of the data, various visualisations have been proposed. The main goal of visualisation is to provide a more intuitive and understandable view on the data map. Besides, plain SOM clusterings do not include visual enhancements of any kind and seem rather unattractive to end users. SOM visualisations can be based on more than one level: the high-dimensional input space and the two-dimensional output space, or combinations thereof. Visualisation based on single components of the input vectors take into account the elements of the input space, the most commonly used method is Component Planes, i.e. the colour-coding of single components. The *Rhythm Patterns* visualisation, for in-



Figure 2. PlaySOM running on a Tablet PC.

stance, is based on colour-coding different groups of components, illustrating different sonic aspects like low frequencies and bass-dominated music. A visualisation technique that shows the local cluster boundaries by depicting pair-wise distances of neighbouring prototype vectors, the unified distance matrix (U-Matrix), is described in [10]. It is the most common method associated with Self-Organising Maps and has also been extended in numerous ways. More sophisticated approaches exploit e.g. the analogy to geography [8] or use an island-like metaphor to visualise important regions of the map [6].

User interfaces based on the SOM have been researched by several teams, yielding in miscellaneous implementations. Examples are MusicMiner [5] or a three-dimensional user interface [2], based on a three-dimensional audio landscape. A game pad is used for navigation through the map and additional information like labelling using web data and album covers are displayed in the three-dimensional space.

The clustering of pieces of audio based on content features as well as metadata attributes using a spring model algorithm is described in [11]. This user interface is particularly developed for small-screen devices.

3 Multimedia Interfaces

Users often face a huge number of songs in their collection, but listen only to a small fraction, namely the songs or artists they know by name. The recommendation of similar songs to query songs is therefore a much desired feature for private users and commercial vendors alike. This problem can be easily overcome by map based interfaces that allow



Figure 3. Mozart's complete works displayed in i-PocketSOM running on a PDA.

the intuitive selection of regions of similar songs.

Therefore map based access to audio collections has several advantages compared to traditional browsing of hierarchies. First, it provides an intuitive way of organising collections, having similar songs on adjacent regions on the map, letting users easily get accustomed to their own or new collections. Secondly, a range of selection models can be applied, allowing for playlist generation in an intuitive way, allowing for easy selection of smooth transitions between musical styles.

3.1 Desktop Interfaces

The basic idea of the multimedia interface presented in this paper is based on the aforementioned SOM clustering and visualisations.

The *PlaySOM* application is an easy-to-use desktop interface built on top of musical maps to access collections of audio tracks. The main window of *PlaySOM* is the map of music itself allowing the user to select songs for replay. Figure 1 shows the main screen of *PlaySOM* with an example *music map* and a trajectory selection thereon. The playlist window on the left shows the according selection of a smooth transition from 'electronic' music over songs mostly belonging to the genre 'world' back to 'electronic'¹. It also shows a visualisation control element on the lower

¹Please note that the playlist in Figure 1 is displayed in a scrollable list that is partially hidden.



(a) *e-PocketSOM* application run- (b) *m-PocketSOM* on a Sony-ning on a BenQ P50 smartphone. Ericsson mobile phone.

Figure 4. Two versions of PocketSOM running on different mobile devices.

left of the screenshot and settings to change the current visualisation and export/import functions in the menu bar of the application. To gain a more detailed view, the user can use the semantic zooming feature providing different amounts of contextual information according to the zoom level to more closely inspect the songs linked to a certain unit. It supports different modes for selecting songs:

- **Region selection:** provides a square-shaped selection tool enabling users to select the songs belonging to units inside that rectangle without preserving any order of the selected tracks.
- **Line selection:** allows users to draw trajectories and select all songs belonging to units beneath, taking into account transitions between different types of music along the trajectory.

The resulting playlist is shown on the left-hand side of the user interface. Thereby, users can further edit the list before sending it to a music player.

Due to its portable Java implementation the *PlaySOM* application can easily be used on other devices like tablet PCs, as shown in Figure 2.

3.2 Mobile Interfaces

For mobile devices, interfaces have to fulfil special requirements. In addition to the size of the display itself, in-

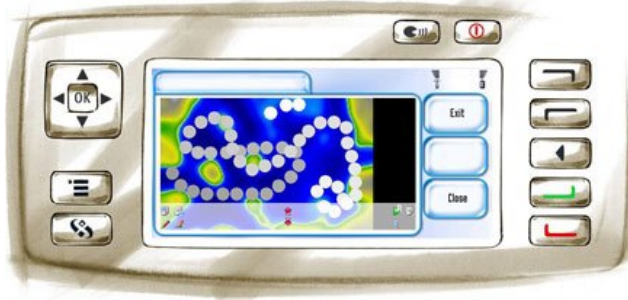


Figure 5. The m-PocketSOM application running on a Nokia 7710 mobile phone emulator.

put possibilities for mobile devices are heavily restricted, comprising (on-screen-)keyboard, touchscreen, character-recognition or special keys. The most difficult question, however, is how to adequately display or gain access to a very large-scale dataset using these minimal interface capabilities. *PocketSOM* is a reduced version of *PlaySOM* application, which provides a simple but intuitive interface. Its functionality focuses on creating playlists using *music maps* generated by the *PlaySOM* application.

We provide three different and independent implementations, each based on a different platform, offering a common set of main features.

- After loading the datafile, the *music map* is displayed.
- Using the touchscreen, the user can now draw a path on the map which will result in a playlist. The songs on this playlist are selected along the path.
- Afterwards the user can manually edit the playlist, such as deleting or reordering items. Finally the playlist is further processed by one of the output modes.

The main interactions with *PocketSOM* such as drawing a path on the *music map* or editing the playlist are done via the touchscreen. Textual input which would require to use the keyboard is either kept to a minimum (e.g. a server address) or assistance will be provided, e.g. through a file open dialog and is saved for as long as the application is running so that these settings have to be set only once.

So far, three different types of output are available:

- *Local Playback*: The playlist generated by the *PocketSOM* can be directly played with a local audio player. Depending on the player, the songs must be available locally or can be streamed via an Internet connection from a remote server.

- *Exporting a Playlist File*: The resulting playlist can be saved to a file, using the M3U playlist format, which can be opened later with an audio player either locally or on another device.
- *Remote Control*: As a third possibility, *PocketSOM* can be used also as a remote control by sending the playlist to a player on a remote device. This player could run on the same host where the music collection is stored on or again stream it over a network connection. VLC² and XPlayer³ are supported in current implementations.

In the following paragraphs we describe differences between the three implementations of *PocketSOM*.

i-PocketSOM Figure 3 shows *i-PocketSOM* running on an iPAQ Pocket PC displaying a *music map* with the complete works of Mozart. This is the first implementation of *PocketSOM* and it is based on Java and SWT (Standard Widget Toolkit). It was developed and tested on an iPAQ Pocket PC.

e-PocketSOM *e-PocketSOM* is an implementation of *PocketSOM* based on *ewe*⁴, a Java-like virtual machine for Windows CE devices. *e-PocketSOM* was developed and tested on a BenQ P50 smart phone, which is shown in Figure 4(a), but is determined to run on any device supported by the *ewe* VM, e.g. PocketPC, MS SmartPhone, Casio BE-300, HandHeldPC Pro, or the Sharp Zaurus.

e-PocketSOM offers settings for, e.g., the target output (M3U file or VLC server) or the selection mode (random or sequential selection). *e-PocketSOM* supports the creation of multiple paths which are afterwards concatenated to a single playlist.

m-PocketSOM Figure 4(b) shows the third implementation of the *PocketSOM* [1]. *m-PocketSOM* is based on JavaME⁵, the Java Micro Edition. Therefore it will run on any mobile device that provides a touch screen, CLDC1.1 (Connected Limited Device Configuration 1.1⁶), MIDP2.0 (Mobile Information Device Profile 2.0⁷) and an implementation of the MMAPi (Mobile Media API⁸) that supports MP3 playback. The application was tested on different mobile phones, including Nokia 7710, as shown in Figure 5, and Sony-Ericsson M600.

²<http://www.videolan.org/vlc/>

³The Xplayer is a simple applet that waits for remote requests from the MapViewer MIDlet (which is part of *m-PocketSOM*) to play songs.

⁴<http://www.ewesoft.com/>

⁵<http://java.sun.com/javame/downloads/index.jsp>

⁶<http://jcp.org/en/jsr/detail?id=139>

⁷<http://jcp.org/en/jsr/detail?id=118>

⁸<http://jcp.org/en/jsr/detail?id=135>



(a) Zooming enabled (b) Playlist editing on the mobile *m-PocketSOM*.

Figure 6. m-PocketSOM.

m-PocketSOM is the only implementation of the *PocketSOM* that supports zooming to provide for more immersive user interaction even on mobile devices, as shown in Figure 6(a)⁹. Again, the generated playlist can afterwards be edited and reordered, as illustrated in Figure 6(b). The resulting playlist can then be either be played locally or sent to the XPlayer running on a remote host.

4 Outlook and Future Work

In this paper we discussed map based user interfaces as alternative ways to accessing audio collections. Clustering was applied and, in connection with visualisation techniques, used to provide another perspective on audio data based on automatically computed acoustic similarity. We further presented a variety of alternative user interfaces to audio collections based on map-like views for various devices ranging from full desktop computers to touchscreen-enabled devices to mobile phones. We showed that simple user interfaces have the potential of being an intuitive entry point to one's private music collection. We also showed that scalability issues can be subdued by the level of abstraction introduced by both the audio feature extraction methods itself and the clustering algorithms applied thereon.

Future work will mainly deal with further investigation of the possibilities of feature extraction and their benchmarking. Further, we plan on extending the range of avail-

⁹Many devices cannot properly handle this feature and therefore it is omitted in the final release, but is still an option for future implementations.

able user interface implementations to other platforms and devices.

References

- [1] P. Hlavac. Innovative user interfaces for accessing music libraries on mobile devices a som based music browser for mobile devices. In *Proceedings of the 7th Workshop on Data Analysis (WDA'06)*. Elfa Academic Press, June 29 - July 2 2006.
- [2] P. Knees, M. Schedl, T. Pohle, and G. Widmer. An Innovative Three-Dimensional User Interface for Exploring Music Collections Enriched with Meta-Information from the Web. In *Proceedings of the ACM Multimedia 2006 (ACMMM'06)*, pages 17–24, Santa Barbara, California, USA, October 23–26 2006.
- [3] T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, Heidelberg, 1995.
- [4] T. Lidy and A. Rauber. Evaluation of feature extractors and psycho-acoustic transformations for music genre classification. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, pages 34–41, London, UK, September 11-15 2005.
- [5] F. Mörchen, A. Ultsch, M. Nöcker, and C. Stamm. Databionic visualization of music collections according to perceptual distance. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, pages 396–403, London, UK, September 11-15 2005.
- [6] E. Pampalk, A. Rauber, and D. Merkl. Using Smoothed Data Histograms for Cluster Visualization in Self-Organizing Maps. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN'02)*, pages 871–876, Madrid, Spain, August 27-30 2002. Springer.
- [7] A. Rauber, E. Pampalk, and D. Merkl. Using psycho-acoustic models and self-organizing maps to create a hierarchical structuring of music by musical styles. In *Proceedings of the 3rd International Symposium on Music Information Retrieval (ISMIR'02)*, pages 71–80, Paris, France, October 13-17 2002.
- [8] A. Skupin. A picture from a thousand words. *Computing in Science and Engineering*, 6(5):84–88, 2004.
- [9] G. Tzanetakis and P. Cook. Marsyas: A framework for audio analysis. *Organized Sound*, 4(30):169–175, 2000.
- [10] A. Ultsch and H. P. Siemon. Kohonen's self-organizing feature maps for exploratory data analysis. In *Proceedings of the International Neural Network Conference (INNC'90)*, pages 305–308. Kluwer, 1990.
- [11] F. Vignoli, R. van Gulik, and H. van de Wetering. Mapping music in the palm of your hand, explore and discover your collection. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR'04)*, pages 409–414, Barcelona, Spain, October 10-14 2004.