SONIXPLORER: COMBINING VISUALIZATION AND AURALIZATION FOR CONTENT-BASED EXPLORATION OF MUSIC COLLECTIONS

Dominik Lübbers
Lehrstuhl Informatik V
RWTH Aachen University
Ahornstr. 55
52072 Aachen, Germany
luebbers@cs.rwth-aachen.de

ABSTRACT

Music can be described best by music. However, current research in the design of user interfaces for the exploration of music collections has mainly focused on visualization aspects ignoring possible benefits from spatialized music playback. We describe our first development steps towards two novel user-interface designs:

The Sonic Radar arranges a fixed number of prototypes resulting from a content-based clustering process in a circle around the user's standpoint. To derive an auralization of the scene, we introduce the concept of an aural *focus of perception* that adapts well-known principles from the visual domain.

The Sonic SOM is based on Kohonen's Self-Organizing Map. It helps the user in understanding the structure of his music collection by positioning titles on a two-dimensional grid according to their high-dimensional similarity. We show how our auralization concept can be adapted to extend this visualization technique and thereby support multimodal navigation.

Keywords: Content-based Music Retrieval, Exploration, Visualization, Auralization, User Interface

1 INTRODUCTION

Ongoing technological advances especially in the field of data compression, storage capacity and network bandwidth have lead to a drastic increase in the size of music collections that are available to today's listener. Online music portals offer their users direct access to an overwhelming number of songs.

To efficiently use this huge amount of music, new access methods have to be developed. These can be roughly categorized into two groups:

1. If the user has a dedicated song in mind and is able to articulate his information demand in some way, well-known music information retrieval techniques can be applied. These include standard database queries on song metadata such as title or artist and content-based queries e.g. following the popular Query-By-Humming application scenario.

2. Sometimes the information retrieval goal cannot be defined precisely. Instead of this, the user might want to *explore* the music collection, take a closer look at pieces that he finds interesting and move around further.

Despite our belief, that the latter paradigm resembles the way customers often behave in music stores, current research in the music information retrieval community has mainly focused on the first approach.

1.1 Related Work

If we restrict ourselves to standard metadata of songs, well-known data visualization techniques developed in the data mining community are applicable. Torrens et al. review three of them (discs, rectangles and tree-maps) in the context of the visualization of personal music libraries [1].

Pampalk calculates an overall song similarity based on perceptual features that mainly model rhythmic aspects of the piece and organizes them on a Self-Organizing Map that preserves the topology of the song space [2]. Titles that are perceptually similar are visualized by so called "Islands of Music".

Despite their difference in the features they use, both approaches rely solely on visual communication between the system and the user. The impedance mismatch resulting from a visual representation of audio information seems unnatural and unnecessary having in mind the human capabilities to process sound information. The well known cocktail party effect can be seen as an example of the powerful audio information processing that our brain is capable of.

Brazil et al. investigate combinations of visual and aural access methods for sound collections [3]: In their first implementation, sound objects are placed on a grid according to selectable properties. The user can navigate this visualization by positioning a cursor on the plane that is surrounded by a circular *aura*: All sound objects that are placed inside this shaded area are played simultaneously and spatialized according to their relative position

to the cursor that models the user's actual position in the sound space.

With the help of the Marsyas audio information retrieval framework the Sonic Browser has been extended to an Audio Retrieval Browser that facilitates the use of content-based features as visualization dimensions [4].

The SoundSpace browser contained in Tzanetakis' audio suite Marsyas3D [5] follows a similar approach: Automatically generated audio thumbnails for music in the neighborhood of the actual selection are played simultaneously. Tailored to its usage in the context of the Princeton Display Wall equipped with a 16-speaker surround system, he is thereby able to realize an intuitive and immersive browsing environment.

Despite these first approaches we believe that more effort is needed to develop content-based multimodal music exploration tools to complement the search-centered activities in the music information retrieval community.

In the next section we briefly discuss two measures to quantify song similarity. This is followed by an introduction to our content-based aural music exploration environment soniXplorer. We present two alternative approaches, namely the Sonic Radar and the Sonic SOM. We proceed with some more technical remarks on our prototype and conclude with ideas for future extensions of this work in progress.

2 SONG SIMILARITY

What makes two songs similar? Similarity is a very high-dimensional measure that can incorporate many different aspects of music, e.g. its melody, harmony, genre, lyrics, etc. Additionally, the notion of similarity is highly user- and context-dependent. So it seems unrealistic to assume a formula that is capable of modelling overall song similarity precisely. Nevertheless, there has been some research to find approximations concentrating on different perspectives.

Pampalk reviewed five different sound similarity models [6] and found out in a simple evaluation, that the similarity measures by Logan et al. and by Aucouturier et al. outperformed the other approaches considered. Both measures concentrate on the spectral characteristics of a song:

Logan and Salomon [7] calculate for each piece a song signature that is basically a weighted set of spectral segment clusters. They compare two signatures using the Earth Mover's distance with a symmetric variant of the Kullback-Leibler divergence as ground distance.

Aucouturier and Pachet [8] represent the "timbral quality" of a song by a Gaussian mixture model (GMM) for the space of MFCCs calculated on short segments. The similarity between two pieces A and B is modelled as the likelihood that A's GMM generates B's feature values. It is approximated using Monte Carlo sampling.

We decided to utilize the approach by Logan and Salomon in our prototype, since its computational complexity is significantly lower. However, this choice can easily be replaced by other alternatives for distance calculation. With the help of classical multidimensional scaling (CMDS) we assign vector coordinates to each piece so that the Euclidean distance between two song vectors resembles the value in the distance matrix. When the number of songs in the collection increases, CMDS might become too time-consuming since its complexity is quadratic in the number of pieces. In this case, the linear FastMap algorithm [9] might be an alternative, although CMDS seems to lead to mappings of higher quality in the context of audio visualization systems [10].

3 SONIC RADAR

Tzantakis et al. mention that only 8 simultaneous audio streams have proven to be practical in the scenario of the 16-speaker Princeton Display Wall [5]. This supports the experience we gained in some initial experiments: Playing back more than two complex songs (like e.g. modern pop/rock-music that are characterized by high loudness values over the whole spectrum) on full level in parallel asks too much from the user's listening capabilities even if the audio streams are spatialized to different stereo-channels: Instead of providing orientation and fast access to the user, he is not even able to recognize one song any more.

Therefore we enhance Brazil's concept of a surrounding aura by introducing a *focus of perception*, that mimics the concentration of our visual perception to the environment of the point we are looking at. We transfer this idea to the audio domain by adjusting the playback level not only according to the distance of the song from the standpoint but also depending on the angle between the beams towards the *focus of perception* and the piece's position (see figure 1). Although song_1 and song_2 share

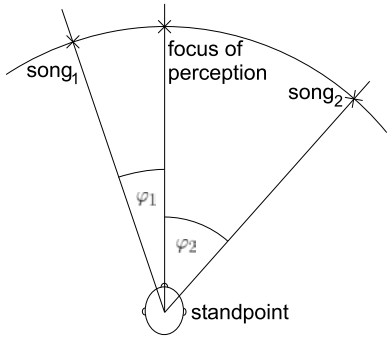

Figure 1: Standpoint and Focus of Perception

the same distance to the user's standpoint, song_1 is played back louder since $|\varphi_1|$ is significantly lower than $|\varphi_2|$. We model this influence on the sound level with a Gaussian density function that is centered at $\varphi = 0°$ and has an user-adjustable variance σ^2. Other windowing functions could be used likewise.

Since we consider a 2-speaker or headphone environment we simply spatialize the mono song according to a linear function depending on φ. If the piece is on the beam between the standpoint and the focus of perception ($\varphi = 0°$), its signal is panned to the left and the right channel equally. For positive φ, the signal is mapped to the left, for negative φ it is mapped to the right channel. The resulting discontinuity at $\varphi = 180°$ has turned out to be no problem since the Gaussian density function for

reasonable φ disappears in this area.

We combine this approach with a simple visualization technique:

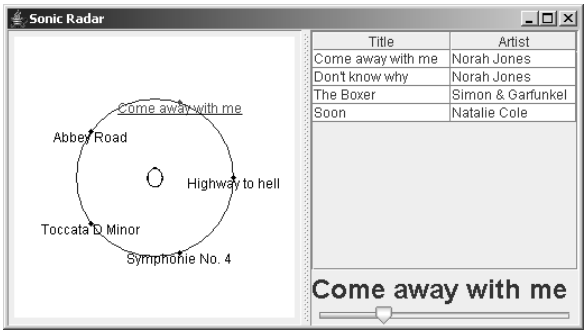

Figure 2: Screenshot of the Sonic Radar prototype

First, the music collection is hierarchically partitioned by successively applying k-means clustering for a user-adjustable number of clusters k. For each cluster a *prototype song* is identified based on the smallest distance to the cluster center. The prototypes of the current subclusters are equally distributed on a circle around the standpoint (see figure 2). This arrangement in conjunction with the directional listening simulation explains the name Sonic Radar for this exploration interface.

The user can rotate the circle to quickly scan through the song prototypes. He can narrow or widen the variance σ^2 to focus on the currently playing piece "in front" or to integrate the surrounding titles in the playback, resp.

As shown in figure 2, the most central prototype is highlighted. Furthermore, all pieces in this cluster are listed on the right hand side of the window. Clicking on one of these titles starts the playback of the song in the media player below. Double-clicking in the Sonic Radar-area allows to step down in the cluster hierarchy and explore the subclusters of the current prototype visually and aurally. Additionally, the user can rearrange the clustered piece by dragging a title from the list and dropping it to a cluster on the left hand side.

4 SONIC SOM

Partitioning a song collection into disjoint clusters we lose much of the similarity information between pieces: The vector space coordinates of a title are reduced to a crisp membership to one cluster; information about the degree of this membership given by the distance to the center and the song's similarity to pieces of other clusters is dropped.

Since humans are used to estimate distances between points on a two-dimensional plane, visualization techniques that map a high-dimensional space to a low-dimensional representation preserving the similarity relationships as far as possible are in widespread use.

One approach based on artificial neural networks proposed by Kohonen [11] is known as the Self-Organizing Map. The (typically 2-dimensional) visualization space is divided into disjoint cells $\{y_i\}$. Each cell is associated with a *weight vector* w_i from the data vector space. In each iteration step t a randomly chosen data point x_j is associated with cell y_{c_j} such that $||x_j - w_{c_j}||$ is minimized. After finding this *Best Matching Unit* the weight vectors of y_{c_j} and its topological neighbors on the map are updated according to the following equation:

$$w_i(t+1) = w_i(t) + \alpha(t) \cdot h_{ic_j}(t)[x_j - w_i(t)]$$

where $\alpha(t)$ denotes the learning rate and h_{ic_j} models the neighborhood relation between cell y_i and the Best Matching Unit y_{c_j}, typically by some Gaussian-like function. As both factors influencing the adaption strength decrease with time t, the SOM converges to a configuration where the Best Matching Units of similar data points are located close to each other on the map.

We apply this sequential training algorithm to calculate a rectangular SOM of 50x50 cells. Depending on the users' current choice of visible area and zoom level this map is converted to a visualization plane of pixels with associated weight vectors that are calculated by bicubic interpolation. The grey value of a pixel is given by the distance of its weight vector to the closest data point. Additionally, a pixel is colored red if its weight vector is the nearest neighbor to a title in the music collection.

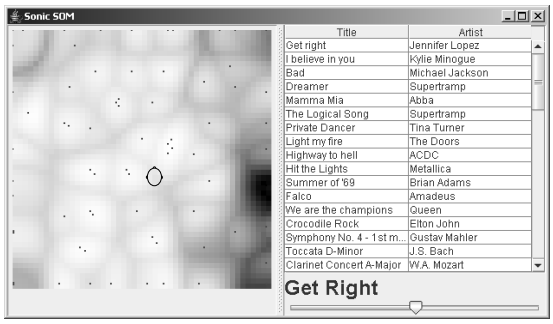

Figure 3: Screenshot of the Sonic SOM prototype

As can be seen in figure 3, this leads to bright areas around clusters of similar titles. The user can change his standpoint in the map, zoom in and out and rotate the SOM plane.

The visual exploration of the SOM is supported by an auralization of the surrounding titles, that resembles the approach described for the Sonic Radar: A focus of perception determines the playback level of the songs in the neighborhood of the standpoint. However, in contrast to the Sonic Radar, pieces can be located very close to each other on the SOM leading to an intransparent mix of different sounds if played simultaneously. Therefore we partition the environment of the standpoint into k disjunct slices and select the closest song in each of these direction classes for playback. Moving to another standpoint, the currently playing pieces are preferably chosen as the new segment prototypes. Thereby we have reduced the SOM auralization problem to a Sonic Radar-like situation.

The currently playing pieces are visualized by blinking pixels. The user can change the current selection of titles by right-clicking on songs to toggle their membership in the playback. Furthermore, the right hand side of the window lists all currently displayed pieces in the collection sorted by their distance from the focus of perception.

5 PROTOTYPE

We are currently developing a prototype to test and evaluate our interfaces. Since the described music similarity measures are readily available in the Matlab toolbox by Pampalk [6], we chose to use it for the calculation of the title distance matrix and the mapping to a vector space by multidimensional scaling.

The Sonic Radar and the Sonic SOM user interfaces are realized as Java applications, that access the precalculated Matlab files. We tested Sun's Java Sound implementation for Windows and found that its playback latency is not acceptable for use in an immersive exploration scenario. To overcome this problem and to be open for extensions to multichannel playback we realized a Java-ASIO-bridge that consists of a thin Java layer communicating via JNI with a C++ layer that handles the function calls and callback hooks to the ASIO interface, for which low-latency drivers even for semi-professional soundcards exist.

For simplicity reasons, the SOM calculation is currently done in Matlab utilizing the SOM Toolbox[1], although we plan to implement it in Java since we consider it to be part of the user interface.

6 CONCLUSIONS AND OUTLOOK

First experiments with our prototype revealed that aural clues indeed improve the user's exploration experience and can help him to navigate the music collection. To find out whether these improvements are significant and which of the proposed interaction concepts is more suitable in which contexts remains to be done in upcoming user studies.

There are still a lot of other open issues we plan to investigate with our prototype:

- How well do the applied algorithms scale for larger music collections?
- What is the effect of reducing the playback to automatically generated song thumbnails as proposed by Tzanetakis [5]? What summarization algorithms are most suitable?
- How can the system be extended to home theater environments with e.g. 5 speakers? How strong is the benefit of such an extension?
- What improvements in the exploration experience can be achieved if not all "activated" songs in the environment sound simultaneously (kind of *time-domain multiplexing*)?
- What improvements can be achieved by emphasizing different frequencies of simultaneously playing songs (kind of *frequency-domain multiplexing*)?

Arranging our test music collection in the Sonic SOM the presented content-based similarity features sometimes resulted in incomprehensible clusters containing pieces that could hardly be judged as similar. To our belief significant improvements in estimating similarity of music can only be achieved by combining different sources of user-adaptive similarity estimates like e.g. presented by Baumann et al. [12].

Strengthening the integration of the user into the IR loop can be seen as a promising way to tackle the complex music information retrieval problem. Our ongoing research on multimodal exploration environments is a step towards this demand for more sophisticated user interaction concepts.

[1] http://www.cis.hut.fi/projects/somtoolbox

REFERENCES

[1] M. Torrens, P. Hertzog, and J.-L. Arcos. Visualizing and Exploring Personal Music Libraries. In *Proc. ISMIR*, pages 421–424, Barcelona, 2004.

[2] E. Pampalk, A. Rauber, and D. Merkl. Content-based Organization and Visualization of Music Archives. In *Proc. ACM Multimedia*, Juan les Pins, France, 2002.

[3] E. Brazil and M. Fernström. Audio Information Browsing with the Sonic Browser. In *Coordinated and Multiple Views In Exploratory Visualization (CMV'03)*, London, 2003.

[4] E. Brazil, M. Fernström, G. Tzanetakis, and P. Cook. Enhancing Sonic Browsing Using Audio Information Retrieval. In *Proc. International Conference on Auditory Display*, Kyoto, Japan, 2002.

[5] G. Tzanetakis and P. Cook. Marsyas3D: A Prototype Audio Browser-Editor Using a Large Scale Immersive Visual and Audio Display. In *Proc. International Conference on Auditory Display*, Espoo, Finland, 2001.

[6] E. Pampalk. A Matlab Toolbox to Compute Music Similarity from Audio. In *Proc. ISMIR*, Barcelona, 2004.

[7] B. Logan and A. Salomon. A Music Similarity Function Based on Signal Analysis. In *Proc. ICME*, 2001.

[8] J.-J. Aucouturier and F. Pachet. Finding Songs That Sound the Same. In *IEEE Workshop on Model based Processing and Coding of Audio (MPCA-2002)*, Leuven, Belgium, 2002.

[9] C. Faloutsos and K.-I. Lin. Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *Proc. ACM SIGMOD*, pages 163–174, 1995.

[10] P. Cano, M. Kaltenbrunner, F. Gouyon, and E. Battle. On the Use of FastMap for Audio Retrieval and Browsing. In *Proc. ISMIR*, 2002.

[11] T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, 3rd edition, 2001.

[12] S. Baumann, T. Pohle, and V. Shankar. Towards a Socio-Cultural Compatibility of MIR Systems. In *Proc. ISMIR*, Barcelona, 2004.