SONGWORDS: EXPLORING MUSIC COLLECTIONS THROUGH LYRICS

Dominikus Baur, Bartholomäus Steinmayr, Andreas Butz
Media Informatics Group
University of Munich (LMU), Munich, Germany
{dominikus.baur,andreas.butz}@ifi.lmu.de, steinmayr@cip.ifi.lmu.de

ABSTRACT

The lyrics of a song are an interesting, yet underused type of symbolic music data. We present SongWords, an application for tabletop computers that allows browsing and exploring a music collection based on its lyrics. SongWords can present the collection in a self-organizing map or sorted along different dimensions. Songs can be ordered by lyrics, user-generated tags or alphabetically by name, which allows exploring simple correlations, e.g., between genres (such as *gospel*) and words (such as *lord*). In this paper, we discuss the design rationale and implementation of SongWords as well as a user study with personal music collections. We found that lyrics indeed enable a different access to music collections and identified some challenges for future lyrics-based interfaces.

1. INTRODUCTION

Lyrics are an important aspect of contemporary popular music. They are often the most representative part of a song. They verbally encode the songs general message, thereby strongly contributing to its mood. For most people, singing along is one of the easiest ways to actively participate in the music experience. Lyrics are also regularly used for identifying a song, since the first or most distinct line of the chorus often also is the song's title. This importance of lyrics makes purely instrumental pieces rather rare in contemporary popular music.

Despite this central role of lyrics, computer interfaces mostly still ignore them. Media player software for personal computers mostly only shows lyrics after installing additional plug-ins, and although the ID3 metadata standard for digital music contains a field for lyrics, it is rarely used. More complex operations, such as browsing and searching based on lyrics, are even further away and scarcely touched in research (e.g., [6]). We therefore think that looking at music from the perspective of lyrics can allow users a fresh view on their collection, reveal unknown connections between otherwise different songs and allow them to discover new patterns between the lyrics and other aspects of the music.

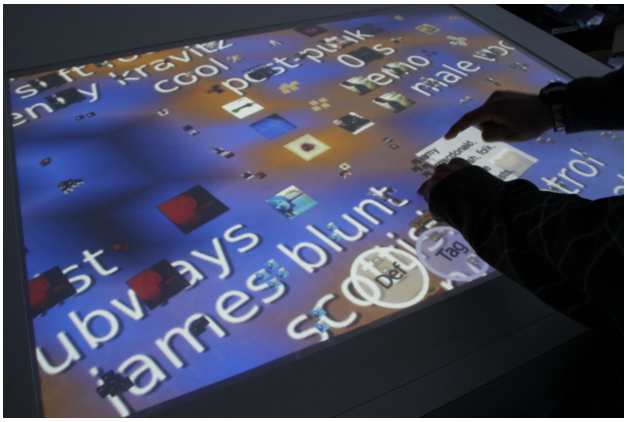

Figure 1. Browsing a music collection through its lyrics on a tabletop

In this paper, we give an overview of SongWords (see figure 1 and video [1]), an application for tabletop computers which supports navigating music collections and investigating correlations based on the lyrics of songs. We present related research on browsing and tabletop interfaces, describe and explain our interface and interaction design decisions, talk about the implementation of SongWords and present the results of a user study.

2. RELATED WORK

Content-based MIR often uses not only the instrumental but also the vocal parts of a song. However, since extracting the words of a song directly from the audio signal has proven to be difficult, a common approach is to gather lyrics from the internet based on available metadata (e.g., [14]). These lyrics then enable tasks that go beyond pure retrieval, such as semantic or morphologic analysis (topic detection [13], rhymes in hip hop lyrics [9], genre classification from rhyme and style features [16]). Other work is approaching the problem of mapping textual lyrics to an audio signal ([12], [7]). Combining an ontology with lyrics enables even more sophisticated tasks: Baumann et al. used natural language processing and mapped text to a vector space model to calculate a lyrical similarity value for pairs of songs [1]. Fujihara et al. presented an approach for creating bi-directional hyperlinks between words in songs that could be applied not only to textual lyrics but also to the actual audio data [6]. They

[1] *http://www.youtube.com/watch?v=FuNPhN6zyRw*

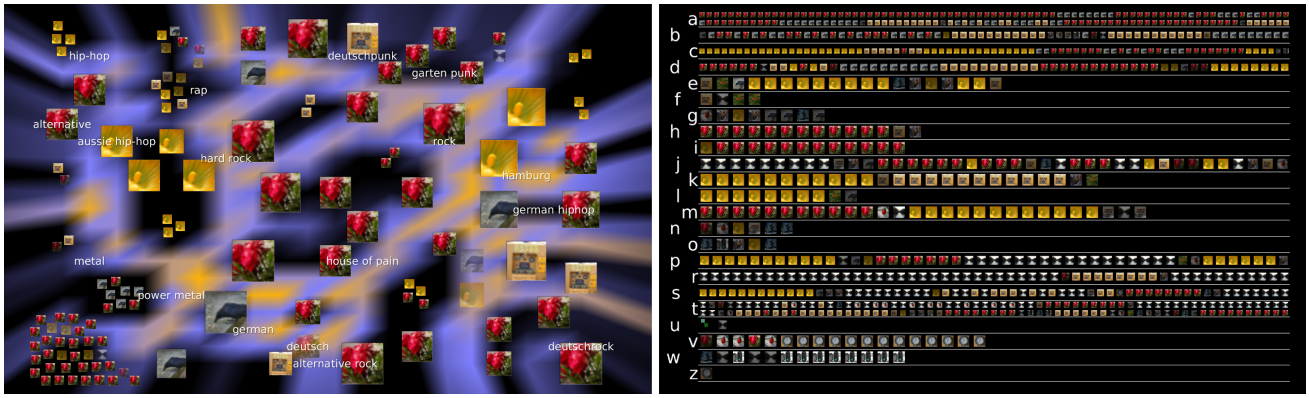

Figure 2. Songs are organized on a map based on lyrics or tags (left), or sorted alphabetically by their artist's name (right)

also describe an application called *LyricSynchronizer* [8] that allows browsing collections by navigating through the aligned song lyrics. There is, however, no work on visualizing a complete music collection based on lyrics.

In order to make complex music collections accessible, a multitude of browsing interfaces are available. Beyond the sorted lists commonly used in media player software, abstraction and filtering capabilities are useful, e.g., by applying techniques from information visualization [23] or by providing views based on different facets [2]. Since music content provides a very high-dimensional data set, complexity also has to be reduced for visualization. Pampalk's Islands of Music [17] is the best known example for this approach. It has also been extended to incorporate multiple views on different acoustic aspects [18]. Self-organizing maps have also widely been used for visualizing text documents (e.g., [5]). In a similar vein, several projects allow browsing a music collection on tabletop displays using self-organizing maps of different low- and high-level audio features (SongExplorer [11], MarGrid [10], DJen [3], MusicTable [22]). Lyrics, however, haven't been used for browsing so far.

3. INTERFACE DESIGN

When designing SongWords we started from two user tasks: First, users should be able to easily browse and search through their personal collections based on lyrics. SongWords should give them a new perspective on their own songs and let them browse through the collection from word to word (similar to [7]). Second, we wanted to allow users to corroborate or disprove hypotheses about connections between lyrics and genres. It should be easy to discover correlations between different genres and words, such as *"Hip hop lyrics often use cuss words"* or *"Pop songs often revolve around 'love' and 'baby'"*.

Since such patterns are hard to discover by scrolling through a text-based list, we decided to map the high-dimensional information space to a two-dimensional canvas using Self-Organizing Maps [15]. Furthermore, as the resulting map at a reasonable level of detail largely exceeded the screen size, we also implemented a Zoomable User Interface to navigate the large virtual canvas on a physical display. With a potentially very large number of items, we finally chose to use an interactive tabletop display for its advantages regarding screen space [24] and its potential for multi-user interaction. In addition, zooming and panning was found to work better using direct touch and bi-manual interaction than using mouse input [4].

3.1 Visualization and Interaction

SongWords analyzes a given music collection and displays it on a two-dimensional canvas. The visualization consists of two self-organizing maps for lyrics and for tags, as well as an alphabetical list by artist's names for direct access to songs (see figure 2). In addition, there is a view for the results of text searches (see below). The user can switch between these different views by pressing one of a number of buttons at the border of the screen.

All songs of the collection are represented on the virtual canvas by their cover art. To optimize the use of screen space, each item is displayed as large as possible without overlapping with other songs. The underlying self-organizing map guarantees spatial proximity between similar items regarding the currently chosen aspect (lyrics or tags). The map contains black areas in the background that connect clusters of items and displays the most relevant words or tags next to the song items to give overview and allow orientation. A common interaction that is possible in each context is pan and zoom (see figure 3). Panning is triggered by putting the finger to the canvas outside of a song icon and dragging, with the canvas sticking to the finger. Zooming and rotation are controlled by two or more fingers and the system calculates the geometric transformation of the canvas from their movements.

In addition to this geometric zoom for the virtual canvas, SongWords also implements a semantic zoom for song icons (see figure 4): At the least detailed zoom level, songs are represented as colored squares to reduce screen clutter with thousands of items. The item's colors represent the home collection of the song when several collections are available. When zooming in, the solid colors are replaced by the artwork of the corresponding record. By zooming further in (or tapping once on the song icon) the artist, title and lyrics of the song become available. Here, the user

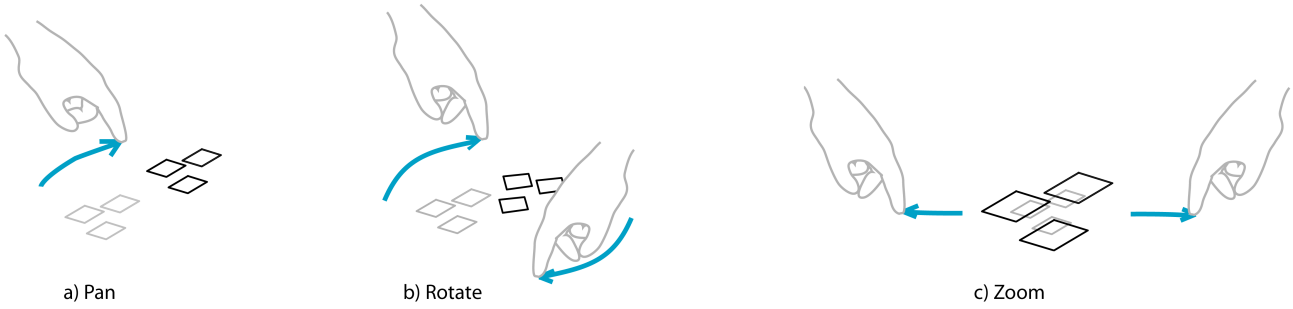

Figure 3. Uni- and bi-manual gestures for panning, rotation and zoom

can scroll through the text if the screen space does not suffice, mark sections of it and search for these sections in the collection. Despite SongWords' focus on text, we decided against using an on-screen keyboard for text search. Not only would it have taken up screen space (or required an explicit mode switch) and suffered from typical problems of virtual keyboards such as missing tactile feedback, it would also have allowed erroneous search terms. Search results in turn are displayed on a spiral based on their relevance (see below). If multiple song collections are visible (e.g., from different users), each song icon has a colored border that represents its home collection.

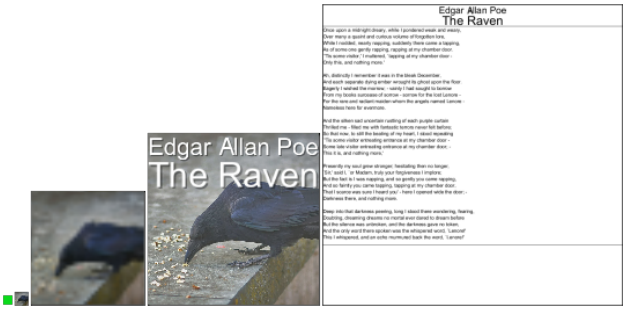

Figure 4. Different semantic zoom levels for song icons

Touching a song with two fingers displays the artist, title and tags for this song. Touching a section of the map with one finger displays the relevant tags or words from the lyrics. Songs can be played by holding a finger on their icon for a short time. In order to allow the discovery of new music based on lyrics, songs with similar lyrics are retrieved from the internet and displayed alongside the songs from the collection. They are rendered slightly transparent in order to distinguish them from local songs. If the user wants to listen to them, a thirty-second sample is downloaded and played.

One challenge in designing for tabletop displays is the so-called orientation problem. While PC screens have a fixed 'up' direction, users can interact with a tabletop computer from any side. The straightforward two-finger rotation of SongWords prevents problems of readability (for a single user) and lets the user quickly change position. When the canvas' orientation changes, the view buttons at the bottom of the screen move along to always remain at the bottom and thus well reachable.

Figure 5. Search results are arranged on a spiral based on their relevance

3.2 User tasks

SongWords enables several distinct user tasks from simple browsing to gaining novel insight and testing hypotheses. By their working principle, the self-organizing maps visualize the similarity between different songs regarding either lyrics or tags. While user-generated keywords can be expected to be relatively consistent for one artist, the lyrics map can bear greater surprises: When songs by one artist are spread widely across the map, this means that this artist produces very diverse lyrics (or employs different songwriters). Similarly, the (in)consistency between songs from different collections can also be seen from their colored borders: If each color is dominant in a different corner of the map, the overlap between the lyrics of the collections is not very high. Discovery of new music based on lyrics is supported in SongWords, as the lyrics and preview clips of related but unknown songs are automatically downloaded and added to fill the map.

The user can navigate from song to song using the text search. By selecting a portion of the lyrics and double-tapping, the system switches to the search view, in which

all songs containing this sequence of words are arranged by relevance. To use the two-dimensional space most efficiently, the linear result list is furled into a spiral (see figure 5). Thereby, the user can quickly find all songs that contain a favorite text section.

To keep track of one or more songs across different views, they can be selected. Songs are selected by pressing the select-button and drawing one or more arbitrary polygons around them. This causes all contained songs to be highlighted with an overlay color, and when switching from view to view their movement can thus be followed effortlessly.

SongWords' different views also allow the user to verify hypotheses. To confirm the hypothesis *Hip hop lyrics often use cuss words* a possible workflow is to switch to the lyrics view, select songs that gather around cuss words, then switch to the tag-view (where the genre of a song is usually the most prominent tag) and see how many songs appear near to the hip hop area. Similarly, other hypotheses regarding the lyrics of one artist (selected from the alphabetical view) or songs containing a certain word can be examined.

4. IMPLEMENTATION

The SongWords prototype has been implemented in C++ for Windows XP. We use the OpenGL framework for rendering and the BASS library for audio playback. SongWords was deployed on a custom-built FTIR tabletop display (99 x 74 cm, 1024 x 768 resolution) and multi-touch input is handled by the Touchlib library. We rely on the internet for all information besides the actual artist, title and album information of a song: After extracting this information from the MP3's ID3 tag, we use the search capabilities of various internet lyrics databases (e.g., lyrics.wikia.com) and parse the resulting HTML pages (similar to, for example, [14]) to retrieve the lyrics. Album covers and user-generated tags are accessed through the API of Last.FM. In order to find the most representative words for a song, we filter the text for stop words and afterwards perform a term frequency inverse document frequency (TF/IDF) analysis [21] to find discriminative terms. The resulting word lists are stemmed with Porter's stemming algorithm [20], to merge related words. These stemmed words are also shown on the map, and in some cases look misspelled. The list of all discriminative stemmed terms forms the feature vector which is used for computing the self-organizing map [15].

For creating the self-organizing map of tags, each song item again receives a feature vector consisting of the tags and their popularity. Our self-organizing maps are based on the classical machine learning approach by Kohonen [15] with one tweak: Since we wanted to make sure that items do not appear at completely different positions in the lyrics and tag views of SongWords, we don't initialize the learning phase of the tag map with random values, but with the results from the lyrics map. Therefore, the chances that identical items appear at similar positions on the two maps are much higher without disturbing the dimensionality reduction capabilities. We also use a relatively low number of 400 iterations for training in order to generate the visualization sufficiently fast.

To allow discovery and fill the map with relevant related music, for every five songs in the collection a random artist is picked, and a related artist is acquired from Last.FM's API beforehand (related artists are calculated based on collaborative filtering of their massive user base). For this artist, a search on Amazon.com is performed and for each resulting song an audio sample is downloaded. SongWords then tries to find lyrics on the aforementioned online databases and once it succeeds, the complete song is added to the training set of the map.

5. USER STUDY

After implementing SongWords, we evaluated it in order to verify whether it properly supported the tasks for which it was designed. As evaluating a complete visualization system is difficult and an active field of research [19], we decided to rely on qualitative user feedback and a small number of participants.

5.1 Study Design

The main objectives of the study were to check usability of the application and identify possible design flaws that could prevent the user from fulfilling the two main tasks. In addition, we wanted to test the application under realistic conditions and therefore asked participants to select a sample of roughly thousand songs from their personal collections. For this set of songs we gathered covers and lyrics before the study and presented the participants with them. As a third aspect of the evaluation we wanted to verify the choice of using a tabletop display compared to a desktop PC. Therefore, we ported SongWords to a PC and mapped the input to the mouse: A left mouse-click was used for panning and selection, a right click for displaying contextual information and the scroll wheel for zooming.

5.2 Study Tasks

The participants were asked to fulfill tasks of increasing complexity to find potential shortcomings of the user interface. Basic tasks were "Play three different songs" or "Choose a song and find out the last line of the lyrics" which could be easily completed using basic interaction techniques. The more complex compound tasks required participants to combine multiple aspects of the application: "Find your favorite song, pick a word from the text that you regard as special and find other songs that contain it" and "Find words that are typical for a certain genre" were two of them. For each task, we let our participants play around with SongWords without previous instructions in order to gather spontaneous feedback and see how self-explanatory the interaction techniques were. If users weren't able to finish the task on their own the experimenter explained how to do it after a few minutes. Initially, all participants worked on and explored the desktop version of SongWords. After they had fulfilled all tasks, they moved on to the

Tabletop version and completed the same tasks again to find the differences between the two setups. We were only interested in the differences they perceived between the two conditions interaction-wise and not quantitative data like the required time, so using the same tasks and a predefined order of desktop and tabletop led to no problems.

Finally, we also wanted to examine the influence of potential multi-user interaction on SongWords: Therefore, we matched our participants to pairs after each of them had worked on their own, displayed their two collections at the same time with color coding and presented them with multi-user tasks. Exemplary tasks were "Find genres that appear in both collections" and "Find the song from the other's collection that you like best". We thereby wanted to identify potential problems in coordinating the access to the interface and in collaboration between the pairs. In the end, the participants were asked to fill out a questionnaire that collected demographic information and their opinions for the desktop and tabletop version of SongWords.

5.3 Participants

We recruited six participants from the undergraduate and graduate students of the University of *[Removed for anonymous submission]* (age: 24-30 years, one female). We supplied them with a small Java application beforehand that allowed them to conveniently choose a thousand songs from their collections, retrieved required meta-data and sent the results to us via e-mail. Only one of the participants was recruited on short notice and was not able to provide us with his song set, so we used a different participant's collection and adapted the tasks accordingly.

5.4 Results

Using both the tabletop version and the desktop version of SongWords showed that the concepts work similarly well on both platforms. Also, the participants mostly were able to transfer their knowledge from one to the other. One notable exception was "hovering" by using two fingers. None of the users figured this out by themselves. We also observed an artifact from desktop computing: Participants kept trying to double click for starting song playback. The change from a map view to the results view after searching often went by unnoticed as the song's text filled the screen and occluded the switch. Additionally, none of the participants actually discovered new music, as the slight transparency of the suggested items obviously wasn't enough to make them stand out. Most of these flaws can easily be fixed. Besides them, we didn't discover any major usability problems.

Finally, we also observed the participants while interacting in pairs with SongWords. Observations were that the color-coding of items from collections worked, even though clear textual or tag distinctions between the collections were not visible. Also as expected from previous research on tabletop displays, participants quickly began taking turns when interacting with the system in order not to get tangled up.

6. DISCUSSION AND LIMITATIONS

One principal limitation of this approach is the fact that it doesn't apply to purely instrumental music. As discussed in the introduction, this is not a very strong limitation for contemporary popular music, but entirely excludes much of the classical music genre, for example. One of the major challenges in working with actual music collections is their size. The hardware-accelerated graphics of SongWords currently produce interactive frame rates for collections of several thousands of songs, but for a practical deployment there are other restrictions: Gathering the lyrics for a song takes several seconds and has to be performed sequentially (in order not to flood the web servers with requests). The time for training the self-organizing map grows linearly with the number of songs and has to happen twice (once for the lyrics, once for the tags) when the program first reads a new collection. Fortunately, the map can be incrementally updated when new songs are added.

The text analysis capabilities of SongWords are currently limited to the most discriminative terms from each song. These most important words can be seen in the maps at first glance and spatial organization is correctly based on these statistical relationships. As the analysis uses the TF/IDF approach [21], it works especially well when the collection is quite dissimilar regarding the words to produce clear distinctions. Subtler differences will go unnoticed, and would require more sophisticated methods from Natural Language Processing.

7. CONCLUSION AND FUTURE WORK

We have presented SongWords, a user interface for browsing music collections based on their lyrics. The visualization using self-organizing maps, combined in a zoomable user interfaces with interactions for searching, marking and reordering, allows a new perspective on music collections. In particular, we observed that users were able to explore correlations between fragments of the lyrics and genre or other user-generated tags. These correlations would be impossible to discover with current list-based interfaces or visualizations purely based on audio data analysis.

In an evaluation we identified a number of minor design flaws of our current prototype, which we will fix in a future version. We will also explore more sophisticated natural language processing and visualization methods, for example involving synonyms and hierarchical clusters of similarity in order to create an even more meaningful similarity measure on lyrics.

8. REFERENCES

[1] S. Baumann and A. Klüter. Super Convenience for Non-Musicians: Querying MP3 and the Semantic Web. In *Proceedings of the International Conference on Music Information Retrieval*, pages 297–298. ISMIR, 2002.

[2] R. Dachselt and M. Frisch. Mambo: a facet-based zoomable music browser. In *Proceedings of the 6th in-*

ternational conference on Mobile and ubiquitous multimedia, pages 110–117. ACM, 2007.

[3] D. Diakopoulos, O. Vallis, J. Hochenbaum, J. Murphy, and A. Kapur. 21st Century electronica: MIR techniques for classification and performance. In *Proc. 10th International Society for Music Information Retrieval Conference (ISMIR 2009)*, pages 465–469. ISMIR, 2009.

[4] Clifton Forlines, Daniel Wigdor, Chia Shen, and Ravin Balakrishnan. Direct-touch vs. mouse input for tabletop displays. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '07*, pages 647–656, New York, New York, USA, 2007. ACM Press.

[5] B. Fortuna, M. Grobelnik, and D. Mladenić. Visualization of text document corpus. In *Informatica*, volume 29, pages 497–502, 2005.

[6] H. Fujihara, M. Goto, and J. Ogata. Hyperlinking Lyrics: A Method for Creating Hyperlinks Between Phrases in Song Lyrics. In *Proc. 9th International Society for Music Information Retrieval Conference (ISMIR '08)*, pages 281–286. ISMIR, 2008.

[7] H Fujihara, M Goto, J Ogata, K Komatani, T Ogata, and H G Okuno. Automatic synchronization between lyrics and music CD recordings based on Viterbi alignment of segregated vocal signals. In *Proceedings of the Eighth IEEE International Symposium on Multimedia (ISM'06)*, pages 257–264, 2006.

[8] Masataka Goto. Active Music Listening Interfaces Based on Signal Processing. In *IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, volume 2007. IEEE, April 2007.

[9] H. Hirjee and D.G. Brown. Automatic Detection of Internal and Imperfect Rhymes in Rap Lyrics. In *Proc. 10th International Society for Music Information Retrieval Conference (ISMIR 2009)*, pages 711–716. ISMIR, 2009.

[10] S Hitchner, J Murdoch, and G Tzanetakis. Music Browsing Using a Tabletop Display. In *Proc. 8th International Conference on Music Information Retrieval (ISMIR'07)*, pages 175–176. ISMIR, 2007.

[11] Carles F. Julià and Sergi Jordà. Songexplorer: a tabletop application for exploring large collections of songs. In *Proc. 10th International Society for Music Information Retrieval Conference (ISMIR '09)*, pages 675–680. ISMIR, 2009.

[12] M.Y. Kan, Y. Wang, D. Iskandar, T.L. Nwe, and A. Shenoy. LyricaAlly: Automatic synchronization of textual lyrics to acoustic music signals. *IEEE Transactions on Audio Speech and Language Processing*, 16(2):338–349, 2008.

[13] F. Kleedorfer, P. Knees, and T. Pohle. Oh oh oh whoah! towards automatic topic detection in song lyrics. In *Proc. 9th International Society for Music Information Retrieval Conference (ISMIR '08)*, pages 287–292. ISMIR, 2008.

[14] P. Knees, M. Schedl, and G. Widmer. Multiple lyrics alignment: Automatic retrieval of song lyrics. In *Proceedings of 6th international conference on music information retrieval (ISMIR 05)*, pages 564–569. ISMIR, 2005.

[15] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[16] R. Mayer, R. Neumayer, and A. Rauber. Rhyme and style features for musical genre classification by song lyrics. In *Proc. 9th International Society for Music Information Retrieval Conference (ISMIR '08)*, pages 337–342. ISMIR, 2008.

[17] E. Pampalk. Islands of music: Analysis, organization, and visualization of music archives. *Journal of the Austrian Soc. for Artificial Intelligence*, 22(4):20–23, 2003.

[18] E. Pampalk, S. Dixon, and G. Widmer. Exploring music collections by browsing different views. *Computer Music Journal*, 28(2):49–62, Juni 2004.

[19] C. Plaisant. The challenge of information visualization evaluation. In *Proceedings of the working conference on Advanced visual interfaces*, pages 109–116. ACM, 2004.

[20] M.F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[21] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[22] Ian Stavness, Jennifer Gluck, Leah Vilhan, and Sidney Fels. The MUSICtable: A Map-Based Ubiquitous System for Social Interaction with a Digital Music Collection. *Lecture Notes in Computer Science*, 3711/2005(Entertainment Computing - ICEC 2005):291–302, Juni 2005.

[23] M. Torrens, P. Hertzog, and J.L. Arcos. Visualizing and exploring personal music libraries. In *Proc. 5th International Conference on Music Information Retrieval (ISMIR '04)*, pages 421–424. ISMIR, 2004.

[24] Stephen Voida, Matthew Tobiasz, Julie Stromer, Petra Isenberg, and Sheelagh Carpendale. Getting Practical with Interactive Tabletop Displays: Designing for Dense Data, Fat Fingers, Diverse Interactions, and Face-to-Face Collaboration. In *Proc. ACM International Conference on Interactive Tabletops and Surfaces (ITS)*, pages 109–116. ACM, 2009.