AudioRadar: A metaphorical visualization for the navigation of large music collections

Otmar Hilliges[1], Phillipp Holzer, Rene Klüber, and Andreas Butz

University of Munich,
Media Informatics,
Amalienstrasse 17,
80333 Munich, Germany
otmar.hilliges@ifi.lmu.de, philipp.holzer@t-online.de,
klueber@cip.ifi.lmu.de, butz@ifi.lmu.de

Abstract. Collections of electronic music are mostly organized according to playlists based on artist names and song titles. Music genres are inherently ambiguous and, to make matters worse, assigned manually by a diverse user community. People tend to organize music based on similarity to other music and based on the music's emotional qualities. Taking this into account, we have designed a music player which derives a set of criteria from the actual music data and then provides a coherent visual metaphor for a similarity-based navigation of the music collection.

1 About Songs, Playlists and Genres

In the January 27, 2006, edition of People magazine, reviewer Chuck Arnold likens new Australian duo The Veronicas to 'such pop-rock princesses as Avril Lavigne, Hilary Duff and Ashlee Simpson.' He goes on to state that '*Everything I'm Not*, a cut off their album *The Secret Life Of...* was produced by frequent Britney Spears collaborator Max Martin and has a chorus that echoes Kelly Clarkson's *Behind These Hazel Eyes*.'

When we talk about music and try to explain its properties to others, we frequently use constructs describing similarity. One reason for this is that it is much easier to imagine how a piece of music might sound if we can relate it to a song we already know. This also makes it easier to decide whether or not we might like a song or record that is being discussed.

Describing music by similarity to other music seems to work quite well and is widely used in the music press. However, state of the art digital music players like iTunes [2], Winamp [18] or XMMS [28] do not take this into account. All of these players organize digital music libraries using meta information about the songs/albums (e.g. artist, title) and/or a limited set of predefined genres.

This works quite well as long as we know most of the songs that occur in a library and we know into what genre a song or artist fits. But this approach has several implicit problems:

1. Genres aren't expressive enough to cover the breadth of an artist's repertoire. Almost no artist would agree that his entire work can be classified into one single category.
2. Genres are too imprecise to guide users through the vast amount of available music (e.g. the iTunes music store categorizes such diverse artists as punk-rockers Anti-Flag and singer/songwriter James Blunt into the genre "Rock").
3. Genres are very little help to users who want to explore and discover new and unknown music libraries, especially if the artist name is unknown or hard to classify into one of the existing categories.
4. Genres, in general, don't match very well with our moods, e.g. a song from the category rock could be a slow and calm ballad or a fast, rough and loud song.

A major reason for these problems is the imprecise nature of the whole genre concept. With this concept, attempts to classify music often fail because of reasons like ambiguities, subjective judgment and marketing interests. In general, there is a conflict between the broad variety of music (and music properties) and the relatively rigid and error-prone classification system. The fact that meta information is stored in ID3 tags [17], which are created and applied by humans, adds to this problem. In real life most ID3 tags are obtained via online databases like Gracenote CDDB or FreeDB, which are created and maintained by a large community of volunteers. This information is very useful in many scenarios (e.g. displaying song title, album and duration), but there is no quality assurance and, in fact, genre information is often incorrect. For music classification it is a problem that the information is *assigned to* the music and not *derived from* the music.

In response to the problems above, we propose a radically different approach for organizing, browsing and listening to digital music, which is based on two main steps:

1. Instead of relying on meta information, we analyze the music itself, derive a number of meaningful descriptive features from it, and organize the music library by the similarity between songs according to these features.
2. Using this analysis we create a graphical representation for all songs in the library based on similarity. Our visualization employs a radar metaphor as a coherent conceptual model, where similar songs are grouped close together, and the user navigates a musical seascape.

This allows users to surf through their music library (or a music store) guided by similarity instead of scrolling through endless lists.

Our prototype is a new digital music player called AudioRadar. Currently the player has two main functionalities; library browsing and a playlist editor. Both parts of the application are centered around the properties of the actual music and their similarity.

The browser resembles a ship's radar, and the current song is the centroid and similar songs are grouped around it. So a user can immediately understand that nearby songs are similar to the active song but a bit faster/slower or rougher/calmer and so on. The distance from the centroid (along the according dimensions axis) shows how different the songs are.

In the playlist editor users can choose from several dimensions (e.g. speed, rhythm, tone) and specify a range of values she wants to have in her playlist. Thus users can effectively create playlists that suit their mood. This allows the user to create, for example, a playlist containing songs that are relatively slow and calm.

2 Related Work and Contribution

Two different aspects need to be addressed in our discussion of related work to the AudioRadar system; the extraction of musical features and the visualization of the music collection.

Our claim, that the automatic extraction of features from musical data can improve music browsing, is backed up by a number of projects in the music information retrieval community, and an overview of MIR systems is given in Typke et al. [22]. Classification mechanisms range from Metadata-based via collaborative filtering approaches to purely feature-based approaches. McEnnis et al. [15] present a library for feature extraction from musical data and discuss other similar work. Liu et al. [13] propose a method for mood detection from low level features, and Li and Sleep [12] as well as Brecheisen et al. [7] even derive genres from low level features. The Music Genome Project [27] relies on features entered by human listeners to classify music, but uses a collaborative filtering approach to create coherent playlists. Uitdenbogerd and van Schyndel [23] discuss collaborative filtering approaches for music information retrieval and how they are influenced by different factors. Schedl et al. [20] propose to use the co-occurrence of artists on Web pages as a measure of similarity and derive a degree of prototypicality from the number of occurrences. Berenzweig et al. [5] give an overview of similarity measures and discuss how subjective they are, and Ellis et al. [9] question whether there even is a ground truth with respect to musical similarity, but try to provide a number of viable approximations. We do not claim to make a technical contribution in the actual analysis of music, but rather use known methods for extracting the features used in our visualization.

The second aspect of our work is the actual visualization of the music collection. The Information visualization community has come up with a number of ways to present big data sets interactively. Classical examples are Starfield displays and scatter plots. The Film Finder[1] applies the Starfield concept to a movie database with several thousand entries. The motivation behind this work is exactly the same as ours, namely to browse and navigate a complex and high-dimensional space according to some meaningful criteria. The MusicVis system[6] uses a scatterplot-like display. It arranges songs as grey, green or blue blobs in a plane and determines proximity between them by their co-occurrence in playlists. MusicVis can also create playlists from its database, which represent coherent subsets of the music collection with familiar song sequences.

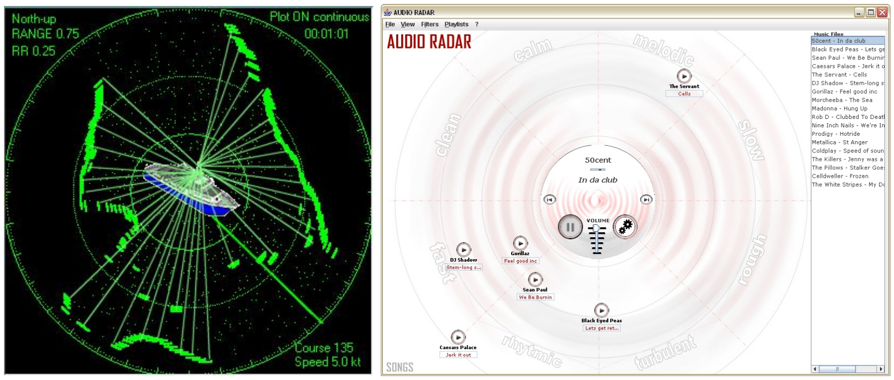


Fig. 1. Left: a real radar screen showing own ship as centroid, green dots symbolize other vessels. Right: the AudioRadar application in browsing mode. The active song is the center point, similar songs are grouped around it.

The Liveplasma Web site [25] presents a graphical interface to a musician and to a movie database. Starting from a search term it presents the closest match in the center of the zoomable display and groups similar artists or movies around it using a spring model based layout mechanism. By clicking on another artist, this one becomes the new centroid and the similar neighbors are dynamically rearranged. Torrens et al. [21] describe visualizations of a personal music collection in the shape of a disc, a rectangle or using a tree map. Vignoli et al. [24, 26] present the artist map, a space-conserving visualization for music collections for use on a PDA screen. Our work goes beyond these existing approaches in providing a coherent mental model, the radar metaphor, for the actual visualization as well as for the navigation of the music collection. Pampalk et al. [19] propose a visualization of feature-based clusters of music as "island of music"; but do not provide a means for navigating this seascape.

Our main contribution over these existing visualizations of music collections is the provision of a metaphor from the physical world. Most of us have an intuitive understanding of what a radar is and how it works. We understand the spatial mapping, which tells us where different objects around us are, particularly in which direction and how far away. This spatial mapping is perfectly applied in our interface, since the more similar songs are displayed closer, and the direction tells us, in which aspect they differ. While this is also the case in the film finder, Liveplasma or with the islands of music, the radar metaphor conveys the feeling of literally navigating the musical seascape and supports casual meandering and browsing.

3 Navigating the Sea of Sounds

The name AudioRadar obviously refers to the metaphor of a ship's radar, a system that is used to detect, range and map objects such as aircrafts and other ships. In our application we calculate the distance of songs between each other by analyzing the audio stream. We use this information to position songs on a radar-like map where the current song is the centroid (Figure 1).

The center area of the AudioRadar player shows the active song and some controls known from standard music players (play, pause, loudness, progress). Radiating out from that centroid are similar songs positioned along four axes. The direction of their offset is determined by the dominant difference from the active song. As shown in Figure 1 this means that "Gorillaz - Feel Good Inc." is faster than "50 Cent - In Da Club". The distance from the center symbolizes the difference in similarity. A song on the outer rim of the radar could be, say, 100% faster then the centroid. The concentric circles in the background function as visual aides to help users judge the distance of two songs.

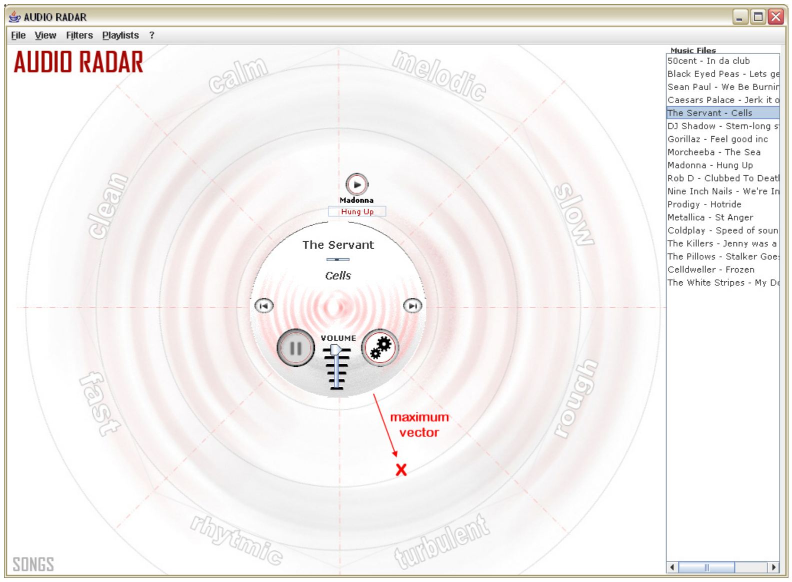
By double clicking one of the songs that appear on the radar (or one song from the list on the right) the user can assign the respective song to become the new centroid. The other songs are relocated according to their similarity toward the new centroid. Each of the similar songs further offers a quick-play option that enables the user to just listen to that song without changing the current setup of the songs.

We experimented with different strategies to position the secondary songs. First we calculated the mean value of all extracted attributes and placed the songs accordingly. In some cases this leads to misleading and even wrong placements (see Figure 2 (a)). For example a song that is more turbulent than the centroid could end up in the melodic sector of the radar because the slow and melodic attributes had high values as well. But it was our intention to create a design that contains all attribute dimensions at once and still allows the user to comprehend the most significant type of similarity at fist glance.

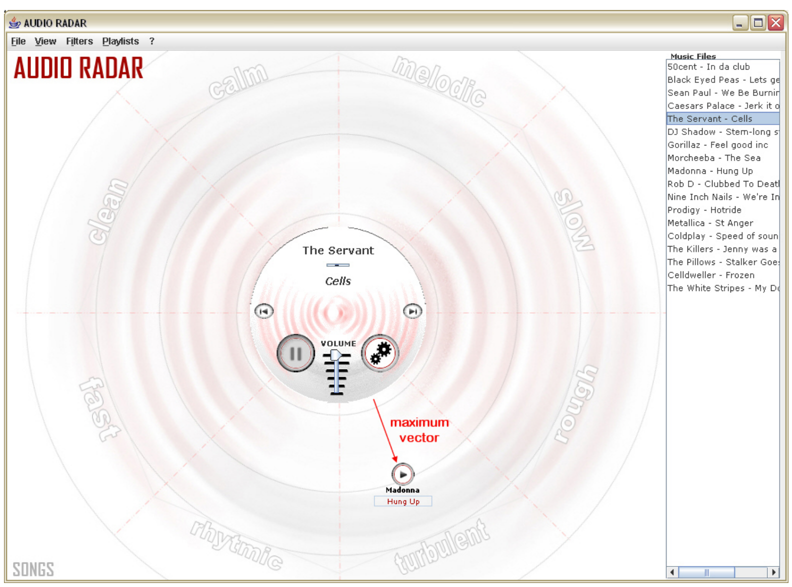
One solution for this problem is to dispose all values but the maximum (see Figure 2 (b)). Thus the placement becomes more coherent with the idea that a song is similar to the current one but only faster, for example. This can lead to visual clutter because songs are only placed on the axes of the radar screen. To avoid this problem we use the second highest value to compute an offset from the axes so that the songs get distributed within the maximum sector (see Figure 2 (c)). Utilizing the second highest value as offset in addition makes the offset meaningful for the user.

3.1 Automatic Audio Analysis

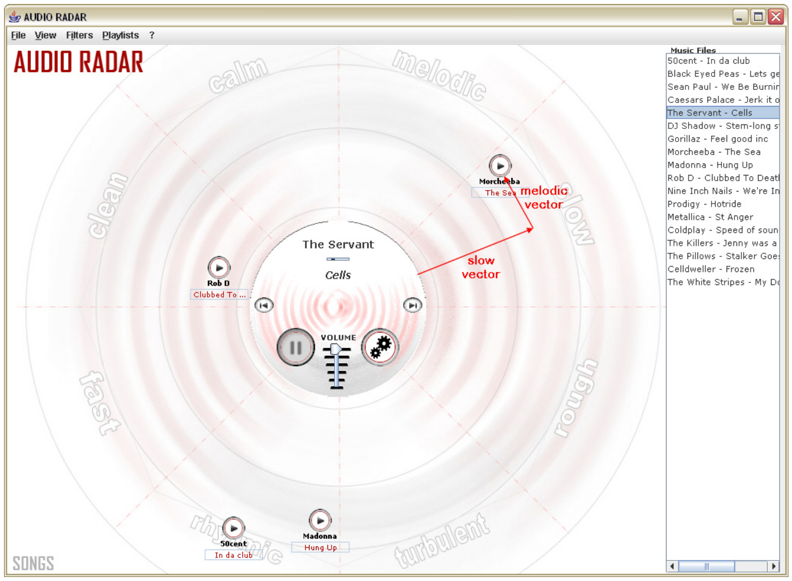
To obtain the data for the placement of each song we analyze the actual audio stream. The four extracted attributes describe each song's position in a four-dimensional feature space. The dimensions are slow vs. fast, clean vs. rough, calm vs. turbulent and melodic vs. rhythmic (see Figure 3). This four-dimensional space is projected onto the two-dimensional display by selecting two of the four dimensions and ignoring the other two (see figure 4). Since the main focus of our work is on the visualization, we used a given analysis library [16] to derive these features. The current results are mostly plausible, but as better algorithms for analysis become available, these can be exchanged in a modular way.


AUDIO RADAR
File View Filters Playlists ?
AUDIO RADAR
calm
melodic
clean
slow
fast
rough
rhytmic
turbulent
Madonna
Hung Up
The Servant
Cells
VOLUME
maximum
vector
SONGS
Music Files
50cent - In da club
The Servant - Cells
Gorillaz - Feel good inc
Morcheeba - The Sea
Madonna - Hung Up
Prodigy - Hotride
Metallica - St Anger
Celldweller - Frozen

(a)


AUDIO RADAR
File View Filters Playlists ?
AUDIO RADAR
calm
melodic
clean
slow
fast
rough
rhytmic
turbulent
The Servant
Cells
VOLUME
maximum
vector
Madonna
Hung Up
SONGS
Music Files
50cent - In da club
The Servant - Cells
Gorillaz - Feel good inc
Morcheeba - The Sea
Madonna - Hung Up
Prodigy - Hotride
Metallica - St Anger
Celldweller - Frozen

(b)



(c)

Fig. 2. Song placement strategies in AudioRadar. a) A misplaced song positioned with mean value placement. b) The same song positioned utilizing only the maximum attribute. c) Preventing visual clutter along axes using second highest value to calculate offset.

The first attribute we extract is the speed of a song. Basically our algorithm counts the beats per minute. However with some songs, especially non electronic ones, we encountered some difficulties with this approach. Finally we modified the algorithm so that it is capable of identifying major repetitive elements and count there occurrence over time.

To determine a song's level of noise we simply consider several intervals of the song and measure the difference between the single intervals. This approach can certainly be improved since we don't take the peculiarities of each song into account. We could, for example, achieve much better results by adjusting the intervals length according to the song's structure (verse, refrain etc.). Another improvement would be to extract continuous elements (baseline, chorus) and specifically consider disharmonies, offbeats and related noise.

The dimension calm vs. turbulent is closely related to the previous one but considers the changes in a song over a greater period of time or, in other words, the amount of differing intervals. Again the same limitations as above apply here.

The last dimension we consider is melodic vs. rhythmic and this dimension is the most problematic one. First of all, our very simple algorithm only extracts very basic information about occurring harmonics and rhythm in the song.

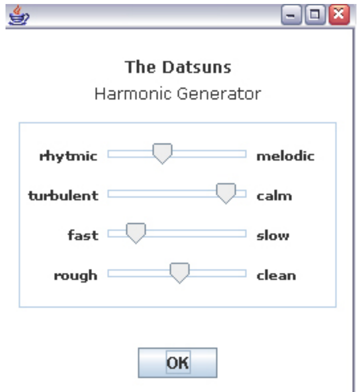


Fig. 3. Results of a song analysis showing the values computed for each of the four dimensions.

Second, this dimension is perceived and judged very subjectively even though it is a very important factor in our liking or disliking off a song. Recent research in the field of music information retrieval has shown that this analysis can be done with quite satisfying results [10, 11]. However, applying state of the art technologies would have gone beyond the scope of this project and remains future work.

3.2 Mood-based Playlist Generation

Most digital music player software is based on the concept of playlists, which provides users with a way to organize playing orders for their songs. Many digital music libraries easily exceed thousands of songs. Because playlists of this size are no longer easy to manage, text-based search functions are provided to retrieve a specific song or artist. Some programs incorporate specific tools to assemble playlists based on meta information [1]. Or we can give up control altogether and use a randomized playing function.

None of those techniques allows users to create playlists based on their current mood. AudioRadar offers such a functionality by letting users define the preferred range of attributes they would like to listen to. With this approach it is not necessary to know all the songs in the library by name (or even at all), and it is not necessary for the user to know what sort of music lays behind a songs name.

The AudioRadar playlist generator gives the user an overview of the complete library (see Figure 4), with the songs represented as small dots. The user now utilizes sliders to specify a range of values for every dimension that she wants to have in her playlist, thus defining a four-dimensional volume. Songs inside this volume are included in the playlist, and songs outside are not. Since it is impossible to render a four dimensional volume into a 2D view we decided to adopt the well know mechanism of color choosers from common painting

[1] e.g. iTunes smart playlists

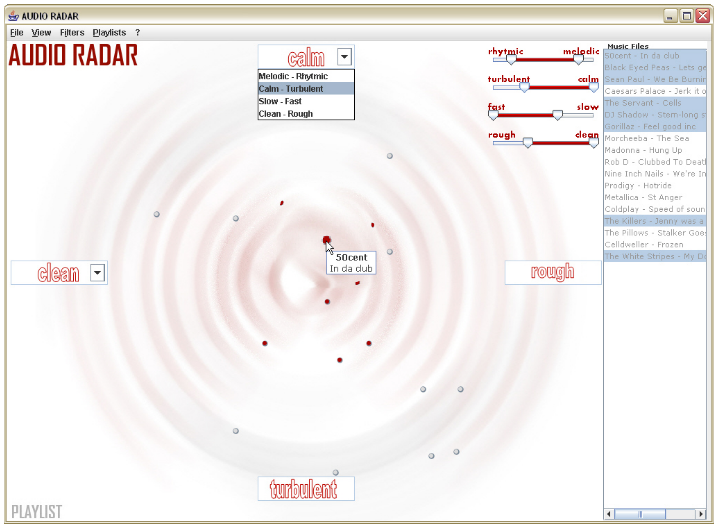


Fig. 4. The AudioRadar playlist creator. A red dot indicates that a song is within the currently selected attribute range while a grey dot signifies that it is excluded from the playlist. Four sliders with two thumbs on the upper right corner of the view control the ranges of the attributes. The list view on the right is linked to the radar view and selected songs are highlighted.

programs where multi dimensional color spaces are mapped onto two dimensions by enabling the user to specify the two displayed dimensions.

In our case this means that each of the four attribute pairs can be mapped to one of the two axes of the radar screen. According to this setting all songs are positioned with their computed values. This view denotes a cut through the 4D volume along the two axes. Any combination of two attribute-pairs can be chosen to determine the allocation of all songs within the view.

Due to the size of digital music libraries, we soon encountered visual clutter to the extent of complete loss of usability. To solve this problem we chose to use very small symbols for the songs and additionally we implemented a fish-eye-lens to magnify the area around the cursor. Hence information about songs can still be retrieved and even songs that are very close to others can be singled out.

4 Conclusion and Future Work

In this paper we have presented a digital music player that supports a new way to 1) browse a digital music library and 2) to generate playlists based on the properties of the music itself and the users mood, not based on names or genres applied to music.

The current status of our work is a fully working prototype that implements all the described functionalities. However it has several limitations that need to be addressed in the future. The most urgent is the quality of the audio analysis which, right now, is still very basic and not accurate enough. Early work in that field has been conducted by Logan et al. [14] and Aucouturier et al. [3] give a good summary of state of the art techniques and their application for music information retrieval. Baumann et al. [4] have shown that the retrieved information can be used very effectivly to increase users music browsing experience.

A more problematic issue is that some aspects important to our liking or disliking of music are very subjective and can't be retrieved from the music itself. We encounter some factors that just can't be measured, such as inspiration, taste or originality. Hence it is very hard to tell whether two songs are perceived as being similar in quality just because they have similar measurable attributes. We simply can't distinguish an uninspired rip-off of a great song by just considering their technical qualities. A solution for this problem might be to consider social and collaborative filtering techniques [8] to incorporate reviews and opinions of music journalists and fans into the rating of songs.

We have not formally evaluated the AudioRadar player, but informal user tests with different target groups, including tech-savvy colleagues, music fans and even musicians, uniformly resulted in very encouraging feedback. In the near future, we plan to improve the audio analysis functionality and conduct a formal evaluation. An especially interesting aspect would be to assess how closely subjective measurements of similarity in songs and the audio analysis of similarity in songs are to one another. Also, we plan to explore in further detail the feature for discovering little- or unknown songs, which a lot of test users especially liked.

5 Acknowledgments

This work has been funded by Deutsche Forschungsgemeinschaft (DFG) and the Bavarian state. We thank Amy Ko for valuable feedback on our manuscript.

References

1. Christopher Ahlberg and Ben Shneiderman. Visual information seeking: Tight coupling of dynamic query filters with starfield displays. In *Human Factors in Computing Systems. Conference Proceedings CHI'94*, pages 313–317, 1994.
2. Apple Computer Inc. iTunes Music player. `http://www.apple.com/itunes`.
3. JJ. Aucouturier and F. Pachet. Music similarity measures: What's the use? In *ISMIR The International Conferences on Music Information Retrieval and Related Activities. Conference Proceedings*, 2002.
4. S. Baumann. Music similarity analysis in a p2p environment. In *WIAMIS '03: Proceedings of the 4th European Workshop on Image Analysis for Multimedia Interactive Services*, 2003.
5. Adam Berenzweig, Beth Logan, Daniel P.W. Ellis, and Brian Whitman. A large-scale evalutation of acoustic and subjective music similarity measures. In *Proceedings of the International Conference on Music Information Retrieval ISMIR 2003*, 2003.

6. Blaine Boman, James Pak, and Sajid Sadi. MusicVis. `http://www1.cs.columbia.edu/~paley/spring03/assignments/HWFINAL/bgb10/`.
7. Stefan Brecheisen, Hans-Peter Kriegel, Peter Kunath, and Alexey Pryakhin. Hierarchical genre classification for large music collections. In *IEEE International Conference on Multimedia & Expo (ICME)*, 2006.
8. William W. Cohen and Wei Fan. Web-collaborative filtering: recommending music by crawling the Web. *Computer Networks (Amsterdam, Netherlands: 1999)*, 33(1–6):685–698, 2000.
9. Daniel P.W. Ellis, Brian Whitman, Adam Berenzweig, and Steve Lawrence. The quest for ground truth in musical artist similarity. In *Proceedings of the International Conference on Music Information Retrieval ISMIR 2002*, 2002.
10. J. Foote, M. Cooper, and U. Nam. Audio Retrieval by Rhythmic Similarity. In *ISMIR The International Conferences on Music Information Retrieval and Related Activities. Conference Proceedings*, 2002.
11. Fabien Gouyon, Simon Dixon, Elias Pampalk, and Gerhard Widmer. Evaluating Rhythmic Descriptors for Musical Genre Classification. 29, 2005.
12. Ming Li and Ronan Sleep. Genre classification via an lz78-based string kernel. In *Proceedings of the International Conference on Music Information Retrieval ISMIR 2005*, 2005.
13. Dan Liu, Lie Lu, and Hong-Jiang Zhang. Automatic mood detection from acoustic music data. In *Proceedings of the International Conference on Music Information Retrieval ISMIR 2003*, 2003.
14. B. Logan and A. Salomon. A music similarity function based on signal analysis. 2001.
15. Daniel McEnnis, Cory McKay, Ichiro Fujinaga, and Philippe Depalle. jaudio: An feature extraction library. In *Proceedings of the International Conference on Music Information Retrieval ISMIR 2005*, 2005.
16. M. McGowan, M. Pfisterer, M. Scheerer, D. Szabo, M. Spears, and P. Santon. JLayer - MP3 Library for the Java Platform. `http://www.javazoom.net/javalayer/about.html`.
17. Nilsson M. Informal standard for ID3 tags version 2.3.0. `http://www.id3.org/id3v2.3.0.html`.
18. Nullsoft. Winamp homepage. `http://www.winamp.com`.
19. Elias Pampalk, Simon Dixon, and Gerhard Widmer. Exploring music collections by browsing different views. In *Proceedings of the International Conference on Music Information Retrieval ISMIR 2003*, 2003.
20. Markus Schedl, Peter Knees, and Gerhard Widmer. Discovering and visualizing prototypical artists by web-based co-occurrence analysis. In *Proceedings of the International Conference on Music Information Retrieval ISMIR 2005*, 2005.
21. Marc Torrens, Patrick Hertzog, and Josep Lluis Arcos. Visualizing and exploring personal music libraries. In *Proceedings of the International Conference on Music Information Retrieval ISMIR 2004*, 2004.
22. Rainer Typke, Frans Wiering, and Remco C. Veltkamp. A survey of music information retrieval systems. In *Proceedings of the International Conference on Music Information Retrieval ISMIR 2005*, 2005.
23. Alexandra Uitdenbogerd and Ron van Schyndel. A review of factors affecting music recommender success. In *Proceedings of the International Conference on Music Information Retrieval ISMIR 2002*, 2002.
24. Rob van Gulik and Fabio Vignoli. Visual playlist generation on the artist map. In *Proceedings of the International Conference on Music Information Retrieval ISMIR 2005*, 2005.

25. Frederic Vavrille. liveplasma. http://www.liveplasma.com/.
26. Fabio Vignoli, Rob van Gulik, and Huub van de Wetering. Mapping music in the palm of your hand, explore and discover your collection. In *Proceedings of the International Conference on Music Information Retrieval ISMIR 2004*, 2004.
27. Tim Westergren. The music genome project. http://www.pandora.com/.
28. X Multi Media System. XMMS homepage. http://www.xmms.org.