MusicRainbow: A New User Interface to Discover Artists Using Audio-based Similarity and Web-based Labeling

Elias Pampalk and Masataka Goto
National Institute of Advanced Industrial Science and Technology (AIST)
IT, AIST, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan

Abstract

In this paper we present *MusicRainbow* which is a simple interface for discovering artists where colors encode different types of music. MusicRainbow is based on a new audio-based approach to compute artist similarity. This approach scores 15 percentage points higher in a genre classification task than the similarity computed on track level. Using a traveling salesman algorithm, similar artists are mapped near each other on a circular rainbow. Furthermore, we present a new approach of combining this audio-based information with information from the web. In particular, we label the rainbow and summarize the artists with words extracted from web pages related to the artists. We use different vocabularies for different hierarchical levels and heuristics to select the most descriptive labels. We conclude with a discussion of the results. The first impressions are very promising.

1. Introduction

Declining production costs have made it much easier for artists to produce their own music without the support of record labels. Furthermore, the Internet enables artists to easily reach out to their audience. However, at the same time the competition for attention has become harder. Music listeners are confronted with an abundance of choices. The work presented in this paper aims at supporting listeners in discovering artists they might not discover otherwise.

There are many ways to discover artists, including, for example, reading reviews and recommendations, browsing lists of similar artists (e.g. amazon.com or allmusic.com), participating in community networks (e.g. myspace.com), or listening to personalized Internet radio (e.g. last.fm).

In contrast to these existing approaches, we use content-based analysis to compute artist similarity. In particular, we do not use collaborative filtering or manually annotated data. Content-based approaches have the advantage that they are not biased by popularity (unlike collaborative filtering) and are much cheaper than manual annotations. Furthermore, we do not assume that the users have specific artists in mind when they start their exploration. This is in contrast to systems such as liveplasma.com which require the user to enter an artist's name before they recommend similar artists.

Our primary objective is to support users in discovering new artists. Given a large music collection with many different styles of music, the users should be able to quickly get an overview of the contents. The secondary objective is to keep the interface as simple as possible. Usage of the interface should be intuitive and require no training. The interface should be implementable on mobile devices with limited screen sizes including phones and music players.

In this paper we present a new music interface to discover artists. Artists are arranged on a circular rainbow (with no beginning or end, see Figure 1). Similar artists are located close to each other on the rainbow. The similarity is computed by analyzing the audio signals. The rainbow is labeled automatically with words occurring on web pages related to the artists. The web pages are found using Google and filtered using specific vocabularies. The rainbow consists of 8 concentric rings having different colors from purple (inside) to red (outside). Each color represents a different style of music. If a color shines brighter in a segment of the rainbow then this indicates that the respective style of music can be found there. Furthermore, we apply audio summarization techniques so that the users can quickly get an overview of each artist.

As input device we use a Griffin Powermate knob (see Figure 1). Alternatively, similar interfaces (such as the wheel used for iPods) could be used. The user interacts with the visualization by rotating the rainbow (by turning the knob) and selecting an artist to listen to (by pushing the knob).

2. Related Work

In [1] an approach was presented with the same objectives and using similar techniques to describe artists and groups of artists with words. In contrast to this paper, the artists were organized using a hierarchical tree structure, and neither audio-based analysis nor visualizations were used.

In [2] a similar approach was used to map music to a circle and a wheel was proposed as input device. One of the differences is that we do not aim at playlist generation. In particular, instead of mapping tracks individually, we group them by artists. Furthermore, we use a new visualization metaphor, describe artists using words and audio thumbnails, and label the sections of the circle to help users find interesting sections more easily.

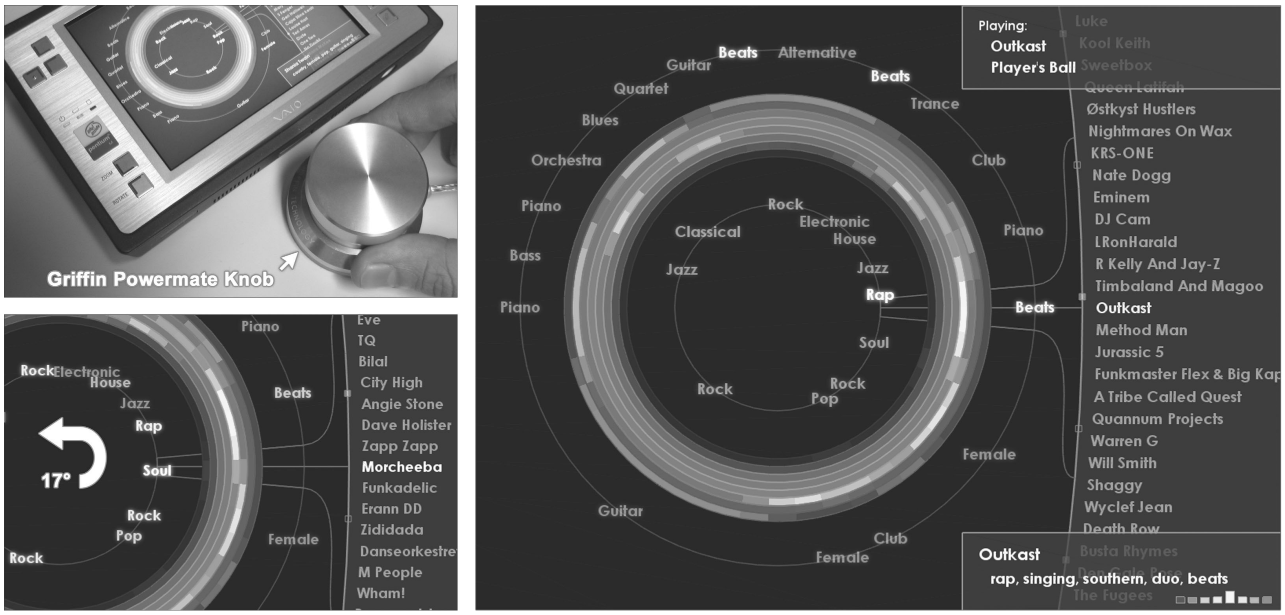

Figure 1. The MusicRainbow interface. The input device (Griffin Powermate knob) is shown in the upper left. On the right side is a screenshot where Outkast is selected. The picture in the lower left shows how the colors and labels change when the rainbow (shown on the right) is rotated counter-clockwise by 17 degrees.

Overall, there are a number of related approaches. Related user interfaces include, for example, the work presented in [7] which helps users discover and enjoy music using audio-based similarity. Approaches with a stronger focus on visualizations (e.g. using maps to display the contents of a collection) include, for example, the work presented in [3–5]. Approaches without a particular focus on visualization include, for example, work using user profiles and information on the web to make recommendations [6]. Furthermore, related work includes automatic playlist generation (e.g. [8–12]).

3. User Interface

Figure 1 shows the MusicRainbow interface. Inside the circular rainbow are high-level genre terms which describe the different sections (e.g. Rap, Jazz, Soul). More specific terms (e.g. Female, Guitar, Beats) are located outside. Each ring of the rainbow (and thus color) corresponds to one high-level term. In sections where the terms are frequently used to describe the artists, the colors are brighter. For example, in the screenshot one of the yellow-green colors corresponds to Rap and red corresponds to Rock. (However, this information is not given explicitly to the user.)

By turning the knob, the user rotates the rainbow (and labels). There is no linear mapping between positions on the knob and the rainbow. If the knob is turned faster, the rotation step size is larger, allowing the user to quickly reach a different region on the rainbow. If the knob is turned slowly, the step size is small allowing the user to move between adjacent artists on the rainbow.

In the center of the right side is the name of the currently selected artist. In the screenshot (Figure 1, right side), Outkast is selected. Artists in the neighborhood include, for example, Eminem, Jay-Z, Busta Rhymes, and Warren G. At the bottom right, the selected artist is described with some words. For Outkast the following (automatically generated) description is displayed: “rap, singing, southern, duo, beats”. (Note that Outkast is a duo from Atlanta.)

Furthermore, below the description, 8 color bars (corresponding to the colors of the rainbow) are displayed. These color bars help the user understand which colors correspond to which type of music. For example, for Outkast the yellow-green color has the largest bar. This indicates that other segments of the rainbow where this color is highlighted contain similar artists. (In contrast to the rainbow colors, which are smoothed across segments, these color bars are not smoothed. They only describe the respective artists, and thus do not necessarily correspond to the colors of the current segment.)

The labels assigned to the sections of the rainbow are misleading in some cases. For example, a section labeled “Female” can include a large number of male artists. To avoid user frustration, we have implemented two strategies. First, the labels float alongside the rainbow making it less obvious to which segments they apply exactly. Second, the relevant labels for the currently selected artist are always highlighted. This is intended to help the user understand that the labels which are closest to an artist do not necessarily apply. In the case of Outkast, the labels “Rap” and “Beats” are highlighted.

By pushing the knob, a song from the currently selected artist is played. If the user pushes the button again, the next song is played. From each song, only a segment of about 20 seconds length is played. To identify the segments that summarize the song as well as possible, we use a chorus-section detection method (RefraiD [13]). The song continues until the section is over or the user selects a different artist (by turning the knob and pushing it). Pressing the knob for more than 2 seconds stops the music.

All inputs from the user can be given using, for example, a Griffin Powermate knob. There are basically 3 input actions. (1) By turning the knob quickly the user can jump to different regions on the rainbow. (2) By turning the knob slowly the user stays in the same region and can explore similar artists. (3) By pushing the knob the user can select an artist to listen to. The currently playing song is displayed in the upper right. By turning the knob during playback, the user can read the summaries of other artists.

3.1. Implementation and Data

MusicRainbow is built with Processing (processing.org). The computations described in the next section were run in Matlab. For the experiments we used a collection of 15336 tracks from 558 artists. This collection is a subset of the DB-XL collection described in [14].

4. Method

First, we compute the similarities for the artist and map them to the circular rainbow. Second, we mine the web to find words to label the rainbow. Finally, based on these words we compute the colors of the rainbow.

4.1. Mapping Artists to the Rainbow

To compute the similarity of artists, we first compute the similarity of tracks using the approach described in [14] as G1C which is based low-level audio statistics. G1C is a combination of spectral similarity and information extracted from fluctuation patterns.

Given the similarity of tracks, we compute the similarity of artists as follows. First, the similarities are normalized so that for each song the sum of similarities to all other songs equals 1. Second, let $s(t, \mathrm{B})$ be the similarity of track t to the most similar track from artist B. We then compute the similarity $s(\mathrm{A}, \mathrm{B})$ of artists A to B as the average of all $s(t, \mathrm{B})$ over all t belonging to A. Finally, we enforce symmetry by setting $s(\mathrm{A}, \mathrm{B})$ to the minimum of $s(\mathrm{A}, \mathrm{B})$ and $s(\mathrm{B}, \mathrm{A})$.

To evaluate the performance of this approach, we use a genre classification scenario. The assumption is that very similar artists belong to the same genre. We use the DB-XL collection where each artist is assigned to one of 16 genres. (These assignments are only used for this evaluation and not for the interface.) We compute the classification accuracies using leave-one-out cross-validation with a nearest neighbor classifier. On track level, the classification accuracy is 31% (using an artist filter so that training and test set do not contain pieces from the same artist [15]). On artist level, we obtain a significantly better value of 46%. Using web pages (as described below) and the tf$\times$idf similarity implementation described in [1], we obtain about 47% accuracy.

First subjective impressions confirm that the quality of the audio-based similarity is much better at artist level than at track level. We assume the reason for this is that computing similarities for sets of tracks instead of individual tracks is statistically more robust.

Given the distances between artists, we map the artists to the rainbow using a one-dimensional circular self-organizing map [16]. Alternatively any other traveling salesman algorithm could be used (see e.g. [2]).

4.2. Labeling the Rainbow

We query Google via its SOAP interface using the artist's name as exact phrase and "music" and "review" as constraints [17]. We retrieve the top 50 ranked pages per artist and parse them using special vocabularies. For each artist we count how often each word occurs.

We use 3 different vocabularies for labeling and summarizing the artists. One for the labels inside the rainbow, one for those outside, and one for short descriptions of the artists. All vocabularies are subsets of the vocabulary used in [1]. As recommended in [1], some words such as "song" or "group" were removed. Furthermore, the vocabularies contain lists of similar words. For example, "rap", "hip-hop", "hip hop", and "hiphop" are treated as one term.

For the labels on the inside of rainbow, we use a vocabulary of over 50 high-level terms such as "rock". For the labels on the outside, we use terms describing various styles of music (e.g. "synthpunk"), various instruments (e.g. "flute"), and various words commonly associated with music (e.g. "contemporary"). Not included are words from the high-level vocabulary, nor words which we consider to be generally unrelated to the acoustical characteristics of the music (e.g. names of cities or countries). The third vocabulary contains about 1400 words. It contains all words from the other two plus additional ones such as "Berlin".

We use filters to smooth the term frequencies on the rainbow and remove words that occur very frequently (for example, "singing" describes almost half of the rainbow). As a scoring function, we use the technique presented in [18]. In addition, we use simple heuristics to find labels which describe larger regions of the rainbow.

4.3. Coloring the Rainbow

The colors of the rainbow encode the term frequencies of the high-level labels used inside the rainbow. Each of the 8 rings of the rainbow corresponds to a label. The rings are ordered such that the ring representing the most frequent term in the collection is on the outside (red), and the ring representing the least frequent term is on the inside (purple). In Figure 1, the outermost ring corresponds to "rock" which is the most frequent high-level term in the collection. The (orange) ring next to it corresponds to "jazz".

The rainbow is segmented into equally large arcs (note that this segementation is hard to see sometimes because neighboring segments tend to have similar colors). The brightness of each of the 8 colors in a segment is defined by the scoring function used to select the high-level labels. For example, if "rock" has a high average score in a segment then the brightness of the red color is increased.

5. Discussion

We have not conducted a formal user study. However, the first impressions of the interface are promising. The labels on the inside describe the rough structure of the rainbow well. The labels on the outside are useful but not as good as the ones inside. A positive example in Figure 1 is the segment labeled with "female" between the segments labeled with "beats" and "club". The Spice Girls are located in this segment and 9 of their 10 closest neighbors are females (including Jennifer Lopez and Kylie Minogue). However, the number of males in the second "female" segment between "guitar" and "club" is much higher. In particular in the region between "female" and "club", there are less than 50% females.

In general, the quality of the artist summaries seems good. For example, the Spice Girls are described with "pop, female, singing, rock, British". Gilberto Gil is described with "Brazilian, singing, world music, bossa nova". Queen is described with "rock, singing, classic, guitar, opera". The term "classic rock" should have been added to the vocabulary to avoid overlap with "classic music". The term "opera" was probably selected because of their album "A night at the opera".

Errors in the summaries occur most frequently in cases where the artist name is ambiguous (e.g. in the case of "Elisabeth"). An example for a unique artist name that resulted in a suboptimal summary is Jamiroquai. The band is described with "funk, soul, female, light, cowboy". The reason why "female" was selected is probably their song "Cosmic girl". (Female and Girl are treated as the same term.) The term "cowboy" should be removed from the vocabulary. The reason why it has been selected is probably their album "The return of the space cowboy".

The impressions of using the Griffin Powermate knob as input device are good. Turning the rainbow appears to be an intuitive way to navigate in the collection. However, a force feedback system that would give the user additional information when the selection has been changed from one artist to the next would make the interface even more enjoyable. Alternatively, we have been experimenting with a soft clicking sound to inform the user when the selection was changed (and by how much it has changed).

6. Conclusions

In this paper we presented MusicRainbow, a new interface to explore music collections at the artist level. The colors of the rainbow encode different styles of music. Similar artists are located close to each other on the rainbow. We presented a new approach to compute artist similarities using track level information, and a new approach to combine audio-based similarity and information extracted from web pages (using different vocabularies for different hierarchical levels). We believe MusicRainbow can help users make interesting discoveries amidst the rapidly growing number of artists on the market. Future work will include a user study.

Acknowledgments

This research was supported by CrestMuse, CREST, JST.

References

[1] E. Pampalk, A. Flexer, and G. Widmer, "Hierarchical Organization and Description of Music Collections at the Artist Level," in *ECDL*, 2005.

[2] T. Pohle, E. Pampalk, and G. Widmer, "Generating Similarity-Based Playlists Using Traveling Salesman Algorithms," in *DAFx*, 2005.

[3] E. Pampalk, A. Rauber, and D. Merkl, "Content-Based Organization and Visualization of Music Archives," in *ACM Multimedia*, 2002.

[4] R. van Gulik, F. Vignoli, and H. van de Wetering, "Mapping Music In The Palm Of Your Hand, Explore And Discover Your Collection," in *ISMIR*, 2004.

[5] M. Schedl, P. Knees, and G. Widmer, "Discovering and Visualizing Prototypical Artists by Web-Based Co-Occurrence Analysis," in *ISMIR*, 2005.

[6] O. Celma, M. Ramírez, and P. Herrera, "Foafing the music: A music recommendation system based on RSS feeds and user preferences," in *ISMIR*, 2005.

[7] M. Goto and T. Goto, "Musicream: New Music Playback Interface for Streaming, Sticking, Sorting, and Recalling Musical Pieces," in *ISMIR*, 2005.

[8] B. Logan, "Content-Based Playlist Generation: Exploratory Experiments," in *ISMIR*, 2002.

[9] J.-J. Aucouturier and F. Pachet, "Music Similarity Measures: What's the Use?" in *ISMIR*, 2002.

[10] E. Pampalk, T. Pohle, and G. Widmer, "Dynamic Playlist Generation Based on Skipping Behaviour," in *ISMIR*, 2005.

[11] F. Vignoli and S. Pauws, "A Music Retrieval System Based on User-Driven Similarity and its Evaluation," in *ISMIR*, 2005.

[12] M. Mandel, G. Poliner, and D. Ellis, "Support Vector Machine Active Learning for Music Retrieval," *Multimedia Systems*, Springer, 2006.

[13] M. Goto, "A Chorus-Section Detecting Method for Musical Audio Signals," in *ICASSP*, 2003.

[14] E. Pampalk, "Computational Models of Music Similarity and their Application in Music Information Retrieval," Doctoral dissertation, Vienna University of Technology, 2006.

[15] E. Pampalk, A. Flexer, and G. Widmer, "Improvements of Audio-Based Music Similarity and Genre Classification," in *ISMIR*, 2005.

[16] T. Kohonen, *Self-Organizing Maps*, Springer, 2001.

[17] B. Whitman and S. Lawrence, "Inferring Descriptions and Similarity for Music from Community Metadata," in *ICMC*, 2002.

[18] K. Lagus and S. Kaski, "Keyword Selection Method for Characterizing Text Document Maps," in *ICANN*, 1999.