MusicSim: Integrating Audio Analysis and User Feedback in an Interactive Music Browsing UI

Ya-Xi Chen
Media Informatics
University of Munich
Amalienstr. 17, 80333, Munich, Germany
E-mail: *yaxi.chen@ifi.lmu.de*

Andreas Butz
Media Informatics
University of Munich
Amalienstr. 17, 80333, Munich, Germany
E-mail: andreas.butz@ifi.lmu.de

ABSTRACT

In music information retrieval (MIR), there are two main research directions, which are based either on a folder hierarchy and metadata, or on the actual acoustic content. We believe that both content-based and hierarchy-based retrieval have their respective strengths for browsing and organizing music collections, and that the integration of content analysis techniques in metadata-based media UIs can lead to more powerful UIs. In this paper we present a prototype, in which audio analysis techniques and user feedback are integrated into an interactive UI for browsing and organizing large music collections. We also provide visual assistance to support non-visual perception of music. We discussed our system with test users and received encouragement as well as valuable suggestions for future research.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General terms: Performance, Experimentation, Human Factors.

Keywords: Music information retrieval, music browser, audio analysis, user feedback, playlist generation.

INTRODUCTION

With the development of digital technologies, people are changing their way of organizing, browsing and searching for music. As pointed out by Vignoli [2], users are looking for ways to browse or search songs by mood, social activities or acoustic similarity, which was also confirmed by [4, 5]. Bentley et al. [3] claim that users combine music with their social events, which is consistent with the conclusions of Kim and Belkin [1].

IUI'09, February 8–11, 2009, Sanibel Island, Florida, USA.

Vignoli [2] showed that non-expert users have strong difficulties to express their musical preferences in a formal way and that they often change their minds during the search process. Therefore, Frank et al. [3] and Cunningham et al. [5] also suggested that music UIs should integrate searching and browsing seamlessly and offer functionalities, such as query-by-example, which go beyond explicit search, to allow users to find unexpected but acceptable results. Regarding organization, Cunningham et al. [17] suggest that the visualization of an entire collection will be particularly appealing if it supports rearranging groups of songs. This may make it, for example, significantly simpler to create playlists.

Some researchers have already built such visualizations of large music collections, some of them support easy playlist generation. In Islands of music [7], clusters of similar songs were displayed as islands based on a self-organizing map and low-level features. Artist map [9] combines external data (mood) and raw audio data (tempo, year) to form a 2D map. Users can draw a path and all the items in this path will be compiled into a playlist. Torrens [10] offers three different visualizations, a disc, a rectangle and a tree-map. In their system, playlists are created by selecting and combining regions of interest. Musicream [11] offers a similarity-based sticking function: users can select one song, and then other similar songs will be attached to it automatically, thereby forming a playlist.

INTEGRATING AUDIO ANALYSIS AND USER FEEDBACK IN THE UI

One general problem of many of the above-mentioned approaches based on audio content analysis is, that some items will be wrongly clustered into other groups, which is hard to avoid in fully automatic clustering. Another issue is that the user can only browse passively, but has no control over the layout. We argue that active control and user feedback should be introduced, in order to utilize human perception to improve system performance. Our goal is to realize these functionalities by

- Integrating audio analysis in the UI for both browsing and organizing,

- Harnessing user feedback to improve over fully automatic techniques,
- Provide visual assistance to support non-visual perception of music.

In our previous work AudioRadar [12], a fully automatic visualization of music collections was shown, which was based on low level audio features. Now we aim to enrich such a UI by first using audio analysis to automatically structure the music collection, but then allowing manual overrides of this organization by inferring implicit feedback from the user's interaction with the visualization.

Structure of the MusicSim User Interface

Our prototype was implemented using the prefuse toolkit for interactive Information Visualization [13]. MusicSim presents songs clustered by content similarity and user feedback is derived implicitly from mouse operations (see section 2.4).

Figure 1. The user interface of MusicSim.

The UI contains four main panels: a graph view, a genre panel, an overview and a playlist, as shown in Figure 1. The graph view and the overview panels act as coordinated multiple views. The graph view is not limited in size, thereby not limiting the overall number of songs and clusters which can be shown. The user can freely pan and zoom within the graph view, but will always be provided with an overview of the entire graph in the overview panel on the left.

Since the location of each song relative to the cluster center is determined by its similarity to the centroid and other neighboring songs, a dent in the cluster shape visually identifies outliers, which are less similar than other songs in the cluster. These outliers can easily be dragged to other clusters according to the user's better judgment of similarity, or to any other ordering principle user may think of.

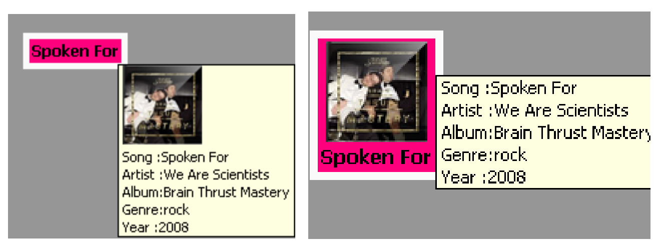
Generation of playlists

One common problem for most projects, which visualized music collections as large maps, was the fact that it becomes difficult for the user to find any specific song. Therefore, MusicSim still offers a text-based search functionality, which can be used to search and locate songs according to a Boolean query the user defines. All matching songs will be highlighted in the graph view and can easily be compiled into the current or a new playlist. All the saved playlists will be shown in the playlist menu.

Earlier work regarding playlists in music retrieval has focused on automatic playlist generation [18, 19]. These systems allowed the user to specify the length of the playlist and define some seed songs. We believe that music similarity is indeed a strong tool for playlist compilation, but that it should just be used to find suggestions, rather than for automatic generation of the entire playlist. In MusicSim, the user can conveniently generate a playlist of similar songs by simply clicking on one or multiple cluster(s), since similar songs are already grouped into one cluster. This is faster than specifying concrete search criteria or manually drawing an explicit path in the UI.

Visual assistance for music perception

One problematic issue about music visualization is the presentation of similarity amongst songs. Music itself carries no visual information, as for example a photo does, which means that the user cannot tell the similarity between songs directly from the UI without playing them one by one. We therefore thought about how to assist the visual perception of music.



a. Text-based b. Cover-art-based

Figure 2. Two different visualizations of a single song.

According to user studies [1, 2], the mental concept of an album is an important concept, and in the searching process, the cover art of the CD offers visual attributes, helping user to find a specific CD. Therefore we offer two visualizations for songs, based on either text or cover-art (see Figure 2). The design goal behind this was to provide a direct visual impression from the cover art. In the browsing process, users can switch between these two modes freely. They can move the mouse over each song to see more detail such as cover art, artist, album and year, or can directly play it by double clicking.

In addition to this, the genre gives more information about what the user can expect from a specific piece of music

[17]. This may even provide a form of social context by allowing to judge the similarity in taste between users. From browsing the genre list of another user's collection, one might, for example, come to the conclusion that this user is much more of a jazz fan than oneself.

In MusicSim, we use color coding for genres. Although there is no commonly accepted matching between musical genres and colors, we have tried to match at least some obvious pairs or use lexical similarity for the color assignment, such as red for rock and blue for blues. Beyond this, we tried to symbolize genres with drastic emotion by brighter colors and peaceful ones by darker colors. Since musical taste is a very subjective issue, this matching is still bound to remain partly arbitrary for different users, and it can therefore be personalized. In the genre panel of Figure 1, the user can see a genre histogram of the entire collection, and unwanted genres can be filtered out by simply deselecting them in the list.

Integration of user feedback

From the discussion above, it becomes clear that it will never be possible to perfectly match a user's understanding of music by a fully automatic approach. One solution to this problem is to take the human into the loop and combine the audio analysis techniques with user feedback. We therefore provide manual overrides of all automatic mechanisms. The simplest such override is to just drag a song from its original cluster to another existing cluster, or to place it in the blank area to create a new cluster. On top of this, there are two ways for reorganizing clusters.

For the manual reorganization, the user can split one existing cluster into two sub-clusters by drawing a line across it (see Figure 3), or merge two clusters by dragging one into each other.

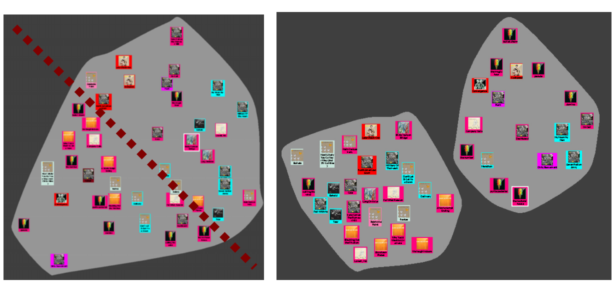


Figure 3. Splitting one cluster into two sub-clusters.

For the automatic reorganization, MusicSim also offers control of the overall layout. If the user wants to change the number of clusters, he/she can simply adjust a slider in the recluster dialog and a dynamic clustering algorithm will reclustered all songs, resulting in a new layout. Figure 4 shows two examples of reclustering results of the same collection.

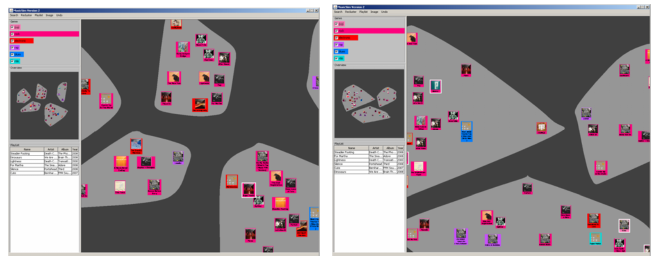


Figure 4. Two different layouts of the same collection.

Since all these operations are executed by the user manually, they can be assumed to be intentional changes and will be saved automatically in real time. Each operation could be undone by global undo functionality.

UNDERLYING AUDIO ANALYSIS

The audio analysis we use, entails several processing steps. First, a set of low level features is extracted from each song. Based on these features, songs are then grouped into different clusters.

Low level features used

A number of open source libraries of audio feature extraction are already available. Four of the representative ones are Marsyas, CLAM, M2K and jAudio. We chose jAudio as our feature extraction package. In order to compute similarity between songs, there are 14 different low level features employed: spectral centroid, spectral rolloff, spectral flux, compactness, spectral variability, root mean square(RMS), fraction of low energy windows, zero crossings, strongest beat, beat sum, strength of strongest beat, MFCC, LPC, method of moments. Phole et al. provide a very good overview [19], which forms the basis for our choice of features.

Clustering algorithm

After extracting these low level features, songs are grouped using a standard clustering algorithm. Since we cannot make any well-grounded assumptions about the size and structure of a music collection in the general case, the number of clusters in particular is unknown beforehand. Since this would have to be known for a supervised clustering process, we can only employ an unsupervised clustering algorithm. In our implementation, we use Simple K-Means [15].

Depending on the actual degree of similarity between and within clusters, this approach might still create too many or too few clusters in the general case, but as discussed above, this setting can be tweaked manually, as shown in Figure 4.

EVALUATION AND IMPLICATIONS

We have shown the first version of MusicSim to 36 participants who participated in a formative study on music organization and playlist generation (more detail about this study would not be further discussed here since it is beyond the focus of this paper). The result shows that 58.33% of these users thought the visualization UI is very useful and

31.11% thought that it was acceptable. Most of the users even showed a higher interest in the concept of smart playlists based on similarity.

Then we further discussed MusicSim with three users, who applied it to their own, hence familiar music collection. The number of songs in these test cases varies between 271 and 789. We introduced the main functionalities of MusicSim, and then let them play with it using both their own and others' collections. The discussions with the participants were quite encouraging. They were very positive about the idea of such a similarity-based interactive UI and one participant mentioned that he would like to use such a UI even more, if it could be plugged into popular music players, such as iTunes. Besides, we also discussed our focusing issues. For example, all of them agreed, that cover art is a useful visual assistance, that displaying the genre by color coding also helps building a better mental model of their own collection, and that it also allowed them to shape a quick impression of the other users' collections.

Based on the discussion with users, two problematic issues became prominent. Sometimes users expressed the feeling of getting lost when navigating in the large graph view by continuous panning and zooming because of the limitation of screen size. This might suggest that visualization of large music collections is suitable to be implemented on a large display, which is one effective way to avoid frequent panning and zooming.

In addition, musical taste is a subjective issue and different people might have an entirely different understanding of the same music. This is also one of the main reasons why there is no standard definition of genre categories. We randomly chose 52 songs from one participant's collection and asked him to allocate one genre for each song from all the genre categories extracted from the songs' metadata. Comparing the user's answer and the actual genre tag, we found that 25% of them were different, which shows a big difference of genre understanding. However, this user still didn't show any concerns about this result. Moreover, users seldom change the genre tags for songs manually. All of this hints at the fact, that genre tags provide much less objective information than generally implied and that they should just be used as additional information besides standard textual information about song title, artist, album, etc. or relatively objective similarity measures. This latter part of the user test was conducted with only three subjects, and hence is not representative. Nevertheless, it shows a positive attitude towards navigation methods based on similarity.

SUMMARY AND FUTURE WORK

In this paper, we explored an effective way to browse and organize potentially large music collections. A prototype named MusicSim was implemented as an interactive UI. In our prototype, we offer textual information, genre color coding and cover art to assist non-visual perception of music. The user also has access to the layout of the whole UI and he or she can manually override the automatic decisions of the system by providing implicit feedback. Although several forms of feedback are provided in our prototype, we still think that the interaction could be improved by allowing the user to organize music in a more natural and flexible way. It might be useful to introduce a machine learning algorithm to let the UI learn from user feedback and thus become smarter. Our future work will address the generation of smart playlists based on more explicit preferences about similarity, such as voice, instrument, etc. Since current music analysis techniques still lack precision, we are also planning to combine social context, for example, online-community-contributed metadata with content analysis in order to achieve more reliable results.

ACKNOWLEDGMENTS

This research was funded by the Chinese Scholarship Council (CSC) and by the German state of Bavaria. We would like to thank the participants of our questionnaire and user test.

REFERENCES

1. Kim, J.Y. and Belkin, N. Categories of Music Description and Search Terms and Phrases Used by Non-Music Experts. Proc. of the 3rd International Conference on Music Information Retrieval (Paris, France, 2002). ISMIR '02.
2. Vignoli F. Digital Music Interaction concepts: a user study. Proc. of the 5th International Conference on Music Information Retrieval (Barcelona, Spain, 2004). ISMIR '04.
3. Frank B., Crysta M., and Gunnar H. Personal vs. commercial content: the similarities between consumer use of photos and music. Proc. of the SIGCHI conference on Human Factors in computing systems (Montréal, Québec, Canada, 2006). CHI '06.
4. Brown, B., Geelhoed, E. and Sellen, A. The Use of Conventional and New Music Media: Implications for Future Technologies. 8th IFIP TC.13 conference on Human-Computer Interaction (Tokyo, Japan, 2001). INTERACT '01.
5. Cunningham, S, Reeves, N., and Britland, M. An Ethnographic Study of Music Information Seeking: Implications for the Design of a Music Digital Library. Proc. of ACM/IEEE-CS joint conference on Digital libraries (Houston, TX, USA, 2003). JCDL '03.
6. Downie, J. S., and Cunningham, S.J. Toward a Theory of Music Information Retrieval Queries: System Design Implications. Proc. of the 3rd International Conference on Music Information Retrieval (Paris, France, 2002). ISMIR '02.
7. Pampalk, E. Islands of Music: Analysis, Organization, and Visualization of Music Archives. Journal of the Austrian Soc. for Artificial Intelligence, 2003.
8. Knees, P., Schedl, M., Pohle, T., et al. An Innovative Three-Dimensional User Interface for Exploring Music Collections Enriched with Meta-Information from the Web. Proc. of the 14th annual ACM international con-

ference on Multimedia (Santa Barbara, CA, USA, 2006).MULTIMEDIA '06.

9. Gulik, R van, Vignoli, F. Visual Playlist Generation on the Artist Map. Proc. of the 6th International Conference on Music Information Retrieval (London, UK, 2002). ISMIR '05.

10. Torrens, M., Hertzog, P., Arcos JL. Visualizing and exploring personal music libraries. Proc. of the 5th International Conference on Music Information Retrieval (Barcelona, Spain, 2004). ISMIR '04.

11. Goto, M., Goto, T. Musicream: New Music Playback Interface for Streaming, Sticking, Sorting, and Recalling Musical Pieces. Proc. of the 6th International Conference on Music Information Retrieval (London, UK, 2002). ISMIR '05.

12. Hilliges, O., Holzer, P., Kluber, R., Butz, A. AudioRadar: A metaphorical visualization for the navigation of large music collections. Proc. of the International Symposium on Smart Graphics (Vancouver, Canada, 2006).

13. Prefuse, interactive information sisualization toolkit: http://www.prefuse.org/.

14. jAudio: Towards a standardized extensible audio music feature extraction. System,Course Paper. http://www.music.mcgill.ca/~cmckay/papers/musictech/jAudio.pdf, McGill University, Canada.

15. Kanungo, T., Mount, D.M., Netanyahu, N.S., et al. Analysis of a simple k-means clustering algorithm. 16th Annual Symposium on Computational Geometry (Hong Kong, China, 2000). SoCG '00. 100-109.

16. Pampalk, E., Rauber, A., Merkl, D. Content-based Organization and Visualization of Music Archives. Proc. of the tenth ACM international conference on Multimedia .MULTIMEDIA '02.

17. Cunningham SJ, Bainbridge D., Falconer A. ‘More of an Art than a Science’: Supporting the Creation of Playlists and Mixes. Proc. of the 7th International Conference on Music Information Retrieval (Victoria, Canada, 2006). ISMIR '06.

18. Pauws S. PATS: Realization and user evaluation of an automatic playlist generator. Proc. of the 3rd International Conference on Music Information Retrieval (Paris, France, 2002). ISMIR '02.

19. Phole T., Pampalk, E., Widmer G. Generating Similarity-based Playlists Using Traveling Salesman Algorithms. Proc. of the 7th International Conference on Digital Audio Effects (Madrid, Spain, 2005). DAFx '05.