VISUAL PLAYLIST GENERATION ON THE ARTIST MAP

Rob van Gulik
Institute of Information and Computing Sciences, Utrecht University
Padualaan 14, 3584 CH, Utrecht (NL)
rob@cs.uu.nl

Fabio Vignoli
Philips Research Laboratories
Prof. Holstlaan 4, 5656 AA Eindhoven (NL)
fabio.vignoli@philips.com

ABSTRACT

This paper describes a visual playlist creation method based on a previously designed visualization technique for large music collections. The method gives users high-level control over the contents of a playlist as well as the progression of songs in it, while minimizing the interaction requirements. An interesting feature of the technique is that it creates playlists that are independent of the underlying music collection, making them highly portable. Future work includes an extensive user evaluation to compare the described method with alternative techniques and to measure its qualities, such as the perceived ease of use and perceived usefulness.

Keywords: Playlist generation, visualization of music collections, artist map.

1 INTRODUCTION

Playlists play an important role in the user experience of dealing with digital music collections, on portable players as well as desktop machines [1]. Unfortunately, creating a playlist is not an easy task. Given the large number of songs in many digital music collections and the short time people are generally willing to spend making a selection, a tradeoff has to be made regarding the quality of the list and the time spent creating it. Often a rough selection is made offline, which is refined during the actual playing of the list. For large collections (i.e. more than 1000 songs), it is not uncommon that either a large part of the music is ignored, or the whole collection is simply played at random.

What makes playlist creation hard is the granularity of item selection: selecting songs one by one can be difficult, because you have to know and remember song titles and artist names. Also, having to select songs in this linear fashion takes a long time. It may be more convenient to select certain *kinds* of music.

One of the main reasons people do not want to spend much time creating a playlist is that these lists go out of date. Especially on portable music players, digital music collections are highly dynamic. New popular songs may replace yesterday's hits on an almost daily basis.

In this paper we propose a visual playlist creation method that requires little interaction, while giving the user substantial high-level control over the selected songs. Moreover, the resulting playlists are independent of the music collection for which they were originally created. The paper is organized as follows: Section 2 gives an overview of the related work. Section 3 describes the playlist creation interface, and in section 4 we discuss the implementation; the actual selection of songs based on the playlist path created by the user. Finally, section 5 presents our conclusions and some directions for future research.

2 RELATED WORK

The visual playlist creation method described in this paper relates to two fields of research; visualization of music collections, and automatic playlist generation.

2.1 Visualization of music collections

The visualization of music collections has been researched at many different sites. Pampalk et. al. created the *Islands of Music* system in which pieces of music are organized by a self organizing map, based on perceived sound similarities [2]. The result depicts clusters of similar music as islands on a geographical map.

The *artist map*, introduced in [3] by van Gulik et. al., also provides an annotated overview of a music collection in the form of a two dimensional map. It uses data from external sources (i.e. mood) in addition to derivatives of the raw audio data (i.e. tempo and timbre) as attributes to organize and cluster music.

Other approaches to visualizing music collections include the audio browser-editor for large wall displays introduced by Tzanetakis and Cook [4], the disc, rectangle and tree-map visualizations by Torrens et. al. [5], and the use of a multidimensional scaling algorithm called *FastMap* to visualize songs on the basis of their similarity, by Cano et. al. [6].

All of these visualizations are aimed at providing the user with an easy to understand overview of a large music collection, and most can also be used to browse through such collections. Torrens et. al. go a step further, suggesting that the visual overview can also be used to edit or create playlists, by selecting and combining regions of music of interest.

2.2 Automatic playlist generation

The automatic playlist generation methods we found described in literature are based on one of the following ideas. They select songs similar to one or a few seed songs that were picked by the user [7], they base the selection of songs on user constraints and preferences [8], or they combine these two methods; using constraint satisfaction to find a seed song on which the playlist can be based [9].

This paper introduces a new way of creating playlists on the visualization of a music collection. With this method, users are able not only to select the type of music of interest (as in [5]), but also to control the overall flow of the playlist with minimal interaction. We expect that this intuitive control over music flow can help improve the resulting quality of automatic playlist generation methods that currently often rely on a small number of seed songs. From a different point of view, drawing playlist paths can be seen as an easy way for specifying musical constraints. These constraints can be further used to generate a playlist.

3 VISUAL PLAYLIST CREATION

We use the artist map we described in [3] as a framework to create a new way of making playlists in a fast and easy manner: by drawing paths and/or specifying regions of interest on top of the visualization. The artist map aims at visualizing a music collection in such a way that:

- A clear overview of an entire music collection or a subset thereof can be given;
- Similarity between artists is used and clearly depicted;
- The attributes *mood*, *genre*, *year* and *tempo* label important positions on the map in order to provide context;
- Navigation of a, possibly unknown, music collection is supported by non-specific or fuzzy criteria

The artist map can visualize (large) music collections on a small screen and was designed to support non-specific music searches (without the need to specify artist or song title).

In the map, artists are positioned based on similarity of their music and attribute data extracted from their songs or obtained from external sources. The attribute information is visualized in the form of magnets, which form an integral part of the user interface as well as the placement algorithm (a force-directed graph drawing algorithm, described in [3]). Figure 1 shows an example in which year and tempo magnets are used, and artists are colored on the tempo attribute. The interface also features zooming functionality, which enables the user to select a subset of the music collection for closer investigation, and a list browser that can be used for traditional searches or to inspect and refine a selection.

The interaction required by the user to create a playlist on the artist map consists of the following tasks, which together form the *path drawing phase*:

- Drawing paths and/or clicking individual points of interest on the map
- Specifying the number of requested songs (or the requested length of the list in minutes)

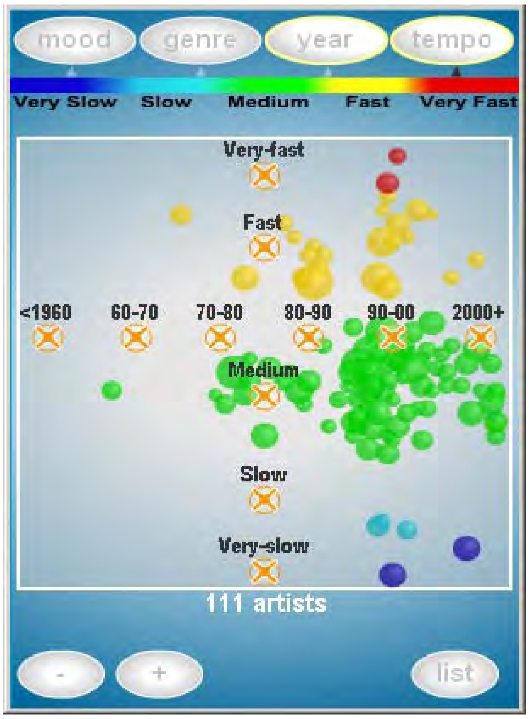

Figure 1. This is an example of a year-tempo artist map. The annotated magnets show that clustering is based on *year of release* along the horizontal direction and *tempo* along the vertical direction, and coloring is done on *tempo*.

Combining this input with the information that is available from the visualization, we end up with reasonable constraints for which a playlist can be generated. The data we use from the visualization are:

- Currently used magnet types (year and tempo in the example shown in Figure 2)
- Current zoom state (upbeat, happy in Figure 2)
- Positions of magnets and artists

In the next section we explain how an actual playlist is constructed using this information.

4 SONG SELECTION METHOD

Selecting songs given the user input and the contextual information is a process that consists of two phases. First the drawn path – consisting of line segments and points – has to be converted to a number of *playlist points*, equal to the requested number of songs. Then, for each playlist point, a song matching the context has to be chosen from the available collection.

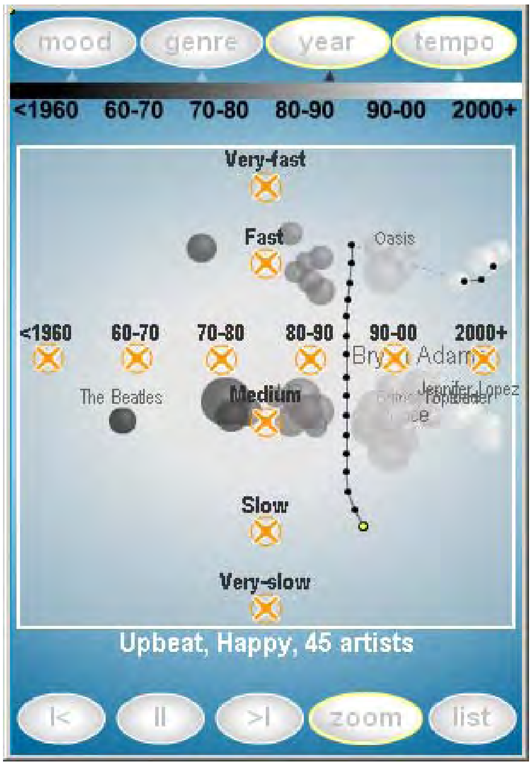

Figure 2. This picture shows an example of drawing playlists on an artist map after zooming in on 45 artists that mainly make happy or upbeat music. Clustering is based on *year of release* and *tempo*, and coloring is done on *year of release*. The playlist starts with music from the 80s and 90s of increasing tempo, after which it will play 3 fast newer songs. All songs will be upbeat or happy, as the user zoomed in on these moods.

4.1 Setting playlist points

A complete playlist path may consist of several smaller parts, in the form of either a path or a point. Informally, the playlist path is the ordered collection of drawings a user has made during the path drawing phase – where a drawing is defined as the recording of mouse movements within the visualization area while the left mouse button is pressed. Each drawing made in this way defines a *subpath*, and each subpath determines a part of the songs in the generated playlist. Figures 3 and 4 each show an example of a playlist path.

Given the playlist path and the requested number of songs *n*, we have to choose n positions on the path from which to play a song. These positions are called *playlist points*. The distribution of playlist points over the subpaths depends on the following priority list:

1. Begin and end points of the complete path
2. Begin and end points of subpaths (if they exist)
3. Remaining points, which are distributed over the sub-paths

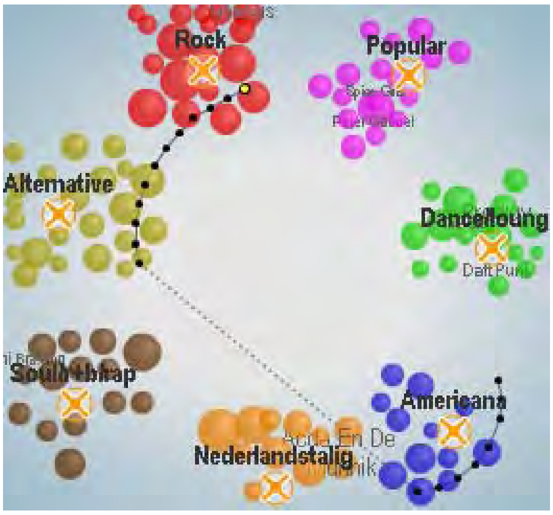

Figure 3. A playlist path created on the artist map which progresses through rock and alternative music, followed by some Americana.

This means that playing a song from both the start and the end of the drawn path is most important. Next, we would like to play at least some songs for each subpath so we put a playlist point at the beginning and end for each of them. For the division of the remaining points, if any, the length of a subpath is used as dividing measure: longer subpaths get more playlist points. We define the ‘length’ of a subpath that consists of only a single point (an individually specified region, see Figure 4) as the mean length of the other subpaths in the drawing. For actually selecting a single song of a certain type, the list interface can be used.

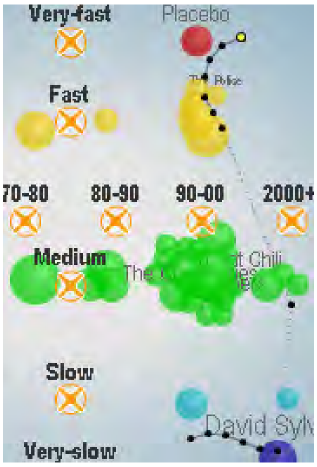

Figure 4. The playlist path shown in this picture starts with fast songs from the 90s, followed by some medium tempo songs from 2000+, and ends with a subpath through slow music.

4.2 Song selection

For each of the playlist points, a matching song has to be found. Song selection is based on the following constraints:

- a song may only occur once in the playlist
- each selected song has to conform to the given zoom-state
- the song selected at each point should reflect its context on the map

Conforming to the zoom-state means that for each magnet type on which the user zoomed in, every selected song should match one of the elements corresponding to the zoom. For example, if the user zoomed in at fast, happy and upbeat, every song in the playlist should be fast, and every song should be either happy or upbeat.

Reflecting context on the map means that for each of the active magnet types, every selected song should match the closest magnet (or match the range between the two closest magnets). Further, every selected song should preferably match the closest artist on the map.

Sometimes the requirements above cannot be combined; the closest artist may not have a song of the required type. In this case, a song is played from the closest artist that does match the constraints. If there is none, we resort to the artists that are not currently on the map and try to find a matching song there. Only if there is no matching song on the device at all, a song will be played that only conforms to part of the criteria. This strategy implies that for a playlist point that lies right on top of an artist, a song of a different artist may be played. But since our method is concerned with creating a playlist based on kinds of music, this is not a problem. If the user wants to include a specific artist, he can do so in the list-based interface.

5 CONCLUSIONS AND FUTURE WORK

The automatic playlist generation method presented in this paper helps music listeners to create playlists in an easy way and in little time. Users only need to specify the kinds of music they want to hear and visually indicate the progression they would like the music in the playlist to have. Furthermore, our method has the following interesting characteristics:

- It is easy to control of the kind of selected music even if you do not remember names well
- playlist paths are independent of the underlying music collection
- the interface was designed to be used on portable music players

We plan to test the playlist creation method described in this paper extensively, by comparing the method with alternative techniques and evaluating the perceived quality of both the interface and the resulting playlist. Other possible future work includes:

- Instead of selecting songs individually, improve playlist coherence by the constraint that two consecutive songs should preferably be similar
- Add other interesting magnet types to the artist map, based on e.g. length of songs, date last played or number of times played last week
- Add the possibility to start with specific items selected in the list based interface, and generate paths with the selection as additional constraint

Acknowledgements

Part of this research has been funded by the Dutch BSIK/BRICKS Project.

REFERENCES

[1] Vignoli, F., “Digital Music Interaction concepts: a user study,” Proceedings of the 5th Int. Conference on Music Information Retrieval, ISMIR 2004, 2004.

[2] Pampalk, E., Rauber, A., and Merkl, D., “Content-based Organisation and Visualization of Music Archives,” Proceedings of the 10th ACM int. conf. on Multimedia, MULTIMEDIA ’02, 2002.

[3] van Gulik, R., Vignoli, F., and van de Wetering, H., "Mapping music in the palm of your hand, explore and discover your collection," Proceedings of the 5th International Conference on Music Information Retrieval, ISMIR 2004, 2004.

[4] Tzanetakis, G., and Cook, P., “MARSYA3D: A prototype audio browser-editor using a large scale immersive visual and audio display," Proceedings of the 7th International Conference on Auditory Display, ICAD 2001, Helsinki, Finland, 2001.

[5] Torrens, M., Hertzog, P., and Arcos, J.-L., "Visualizing and exploring personal music libraries," Proceedings of the 5th International Conference on Music Information Retrieval, ISMIR 2004, 2004.

[6] Cano, P., Kaltenbrunner, M., Gouyon, F., and Batlle, E., “On the Use of FastMap for Audio Retrieval and Browsing,” Proceedings of the 3rd Int. Conference on Music Information Retrieval, ISMIR 2002, 2002.

[7] Logan, B., “Content-Based Playlist Generation: Exploratory Experiments,” Proc. of the 3rd Int. Conf. on Music Information Retrieval, ISMIR 2002, 2002.

[8] Aucouturier, J.-J., and Pachet, F., “Scaling up music playlist generation,” Proc. of the 3rd IEEE Int. Conf. on Multimedia and Expo, ICME 2002, 2002.

[9] French, J., and Hauver, D., “Flycasting: On the Fly Broadcasting,” Proceedings of the 2nd DELOS Network of Excellence Workshop on Personalization and Recommender Systems in Digital Libraries, DELOS 2001, 2001.