

Enhancing Music Maps

Jakob Frank

Vienna University of Technology, Vienna, Austria
<http://www.ifs.tuwien.ac.at/mir>
frank@ifs.tuwien.ac.at

Abstract. Private as well as commercial music collections keep growing and growing. The increasing number of songs in these repositories pose serious challenges to users. PlaySOM and PocketSOM provide map-based access to large audio collections. They provide a quick overview of the whole collection as well as an in-depth view on specific music styles. Furthermore they support the user while exploring and navigating through the collection and provide quick and intuitive playlist creation. But yet, Music Maps have not revealed their full strength. There are still several issues to be solved, such as the continuing growth of collection or multi-user playlist generation. Questions related to these and other issues will be identified and outlined in this paper.

1 Introduction

The immersive grow of private as well as commercial collection of digital audio files has reached a limit where ordinary meta-data based search and browse is no longer sufficient. Several thousand songs can nowadays be stored on personal computers but also on mobile devices, not to speak of the huge amount of music available on commercial audio portals such as iTunes. This huge amount and variety of music calls for novel approaches for searching, browsing and selecting music. Most recent approaches that go beyond textual search and retrieval rely on user-created data such as tags or require social network data. Both techniques suffer from several weaknesses such as the “cold-start” problem that arises for new files in the system.

A novel approach is the usage of Music Maps which arrange and present music on a map-like interface. Based on sophisticated content analysis techniques Music Maps visualise similarities between audio files. This helps to get an overview over large audio collections and provides intuitive and interactive access to them. This novel approach is promising but yet does not reveal its full strength. There are several issues yet to address such as multi user scenarios and the continuing growth of collections.

The remainder of this paper is structured as follows: Section 2 introduces the technical background required for Music Maps. Section 3 will then present the map-based access to large audio collections while Section 4 shows novel applications and interesting issues yet to solve.

2 Technical Background

Music Maps rely on several calculation methods before they yield in intuitive and easy to use interfaces to large audio collections. The creation is basically divided into two steps, both described in the following two sections. In Section 2.3 an experimental approach to bring these techniques to the end user is presented.

2.1 Analysing Music

The first step to Music Maps is the analysis of the audio collection. Different feature extraction methods can be applied to extract meaningful descriptive data from the audio stream, extracting semantic features from music. These features are then useful for a number of music retrieval applications. Typically, features like loudness, rhythm and timbre (among many others) are extracted by computing the power spectrum of the audio signal to obtain a semantic description of the music content. With these descriptors, classification of music into categories is possible, and also automatic organisation of music collections by similarity (see next subsection). By computing distances between the features of the musical pieces, relations of their acoustic similarity can be derived. Songs having a smaller distance in feature space are highly similar regarding the acoustic and musical aspects described by the features. Thus, with audio features extracted from music a direct retrieval of songs sounding similar to given ones is possible without the need of any manually added meta-data. Moreover, this can be used to automatically generate playlists or help users to explore music libraries more intuitively.

PlaySOM and PocketSOM make use of such an audio feature extractor to create a Music Map. To be more precise a feature extractor extracting Rhythm Patterns and Statistical Spectrum Descriptors is used [2]. These extractors include frequency transformation and psycho-acoustical models, and analyse critical bands and modulation frequencies in order to derive fluctuations and statistical descriptions of frequency bands which the human auditory system is most sensitive to. A Rhythm Pattern comprises modulation strength per modulation frequency (in a range of 0 to 10 Hz) for 24 critical bands. High values for a particular modulation frequency in a number of adjacent bands indicate a specific rhythm in a piece of music. Statistical Spectrum Descriptors are derived by computing several statistical measures from a Bark-scale Sonogram [2]. The resulting features convey information about loudness and timbre and are stored in a feature vector, which is subsequently processed by an algorithm which creates music maps (c.f. Section 2.2). PlaySOM and PocketSOM, however, are not limited to these feature sets and can be extended to use other audio descriptors as well.

2.2 Organizing Music

In order to create a Music Map from the features extracted in the previous step a Self-Organising Map (SOM) is used, organising the music on a rectangular area

in such way that music that sounds similar is located together. A SOM is an unsupervised learning algorithm that is used to project high dimensional data points on a 2-dimensional map [1]. The high dimensional data used as input are the feature vectors extracted from the music signal, as described in Section 2.1.

After analysing the audio files the respective feature vectors are provided to the SOM learning algorithm, which iteratively organises the music on a two-dimensional grid in such a way that similar sounding pieces are grouped close to each other. The algorithm works as follows: The map consists of a definable number of units, which are arranged on a two-dimensional grid. Each of the units is assigned a randomly initialised model vector that has the same dimensionality as the feature vectors. In each learning step a randomly selected feature vector is matched with the closest model vector (winner). An adaptation of the model vector is performed by moving the model vector closer to the feature vector. The neighbours of the winner are adapted as well, yet to a lesser degree than the model vector of the winning unit. This enables a spatial arrangement of the feature vectors such that alike vectors are mapped onto regions close to each other in the grid of the units. Once the learning phase is completed, the feature vector of each music file is mapped to its best-matching unit on the map. By that, similar sounding music is located together, with smooth transitions to other musical styles or genres. Note that the axes of the map have no specific meaning, rather they convey the distances among the music files to each other.

2.3 Web Services

One of the biggest difficulties in Music Information Retrieval is to transfer research results such as feature extraction algorithms from research prototypes to user-friendly and understandable applications. One possible way to tackle this challenge is to use the advantages of the ubiquity of the Internet and provide a web service. Web services are a fine possibility to share feature extraction software easily without giving the details on the implementation out of hands. Furthermore, web services can be integrated into almost every application despite of differences in programming language or execution platform.

Another point is that web services allow to delegate intensive calculations to remote servers, without needing much own resources. Especially on mobile devices, where computational power is still the limiting factor, applications that may otherwise not even be feasible can strongly benefit from web services.

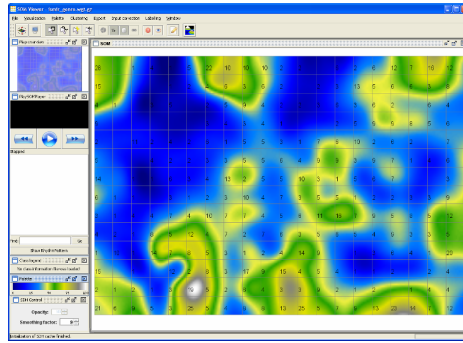
A web service generally consists of two software components: a server providing and a client consuming a specific service. Communication is enabled by the SOAP¹ protocol, which transmits messages in XML format. Our server² currently provides two services: feature extraction from audio and the creation of music maps, though adding more services is easily possible. A demo client implementation that can be used to request the service is also provided.

¹ <http://www.w3.org/TR/soap12>

² The web service, the demo client and all related documents are available under the following URL: <http://www.ifs.tuwien.ac.at/mir/webservice/>

3 Browsing Music Collections

There are many different ways to browse music collections. The most simple is mere directory based browsing while audio player often provide the feature to browse through different hierarchical structures. This is, however, not the best way to explore a audio collection since it does not show relations between songs that go beyond meta-data matching. Both PlaySOM and PocketSOM address this weakness through displaying the similarity different songs by the distance on the map.



(a) The PlaySOM showing a Music Map



(b) PocketSOM on mobile devices

Fig. 1. PlaySOM and PocketSOM

3.1 PlaySOM

The PlaySOM application (see Figure 1(a)) allows users to interact with the Music Map mainly by panning, semantic zooming and selecting of tracks. Users can move across the map, zoom into areas of interest and select songs they want to listen to. It is thus possible to browse collections of a few thousand songs, generating playlists based on track similarity instead of clicking through metadata hierarchies, and listening to those selected playlists. Furthermore it is possible to export them for later use. Users can abstract from albums or genres which often leads to rather monotonous playlists often consisting of complete albums or many songs from one genre. This approach enables users to create playlists based on track not on metadata similarity or manual organisation. By drawing a trajectory on the Music Map it is possible to generate a playlist including smooth transitions between different musical styles. This is especially interesting when browsing very large music collections or when rather long playlists should be generated. Once a user has selected songs and refined the results by manually dropping single songs from the selection, those playlists can be listened to on-the-fly or exported for later use on the desktop machine or even other platforms

like PDAs or Multimedia Jukeboxes if the collection is served via a streaming environment. [3]

Furthermore PlaySOM can act as server in conjunction with PocketSOM providing the Music Map as well as the corresponding audio files for streaming. In this case it receives paths, trajectories and playlists sent by the PocketSOM client to display respective replay them.

3.2 PocketSOM

PocketSOM is a viewer application for Music Maps specially developed and adapted for mobile devices and their limited means of interaction. It allows direct interaction with the map using a touchscreen. This gives intuitive access to large audio collections on small devices. [4]

During the evolvement of PocketSOM several different implementations have been created each specially designed for a specific platform. The most recent and sophisticated implementations are ePocketSOM for Windows Mobile and iSOM for the iPhone/iPod touch (see Figure 1(b)). They are able to load a Music Map over an internet connection from a remote webserver or directly from the PlaySOM application. Furthermore they are able to directly interact with PlaySOM by sending trajectories and paths to be displayed on the map and playlists to be replayed. Finally the above mentioned implementations allow the user full control of the built-in audio player of the PlaySOM application.

These additional connectivity features allow novel applications which will be outlined in the following Section.

4 Future Work

So far, Music Maps on computers and portable devices allow intuitive and interactive access to large music collections. But there are still several issues to solve until Music Maps reveal their full power and benefits.

4.1 Playlist Mapping

The first thing to address is the verification of the path-based playlist generation. The main point is whether user generated “real-world” playlists match the model of trajectories on a Music Map.

So far the assumption is that playlists can be modeled as trajectories on a Music Map. To verify this presumption, user-generated playlists from different sources (e.g. from last.fm³) will be visualised on a Music Map containing the songs used in this playlist along with others from the same style. Then the shape of the resulting trajectory will be analysed. So far, the following shapes are imaginable:

³ <http://last.fm>

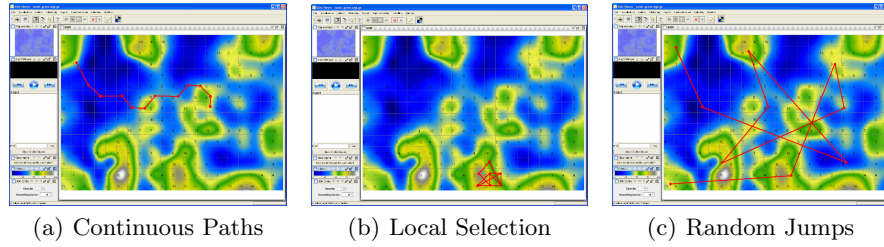


Fig. 2. User-generated playlists mapped on a Music Map.

- **Path:** Playlists do reflect continuous trajectories on the Music Map (Figure 2(a)).
- **local selections:** Playlists stay in a small, isolated area of the Map (Figure 2(b)).
- **random jumps:** Playlists create random long-distance jumps on the Map (Figure 2(c)).
- Any combination of the above mentioned.

Whatever the result of these experiments will be, it will contain valuable information (a) to improve the creation of Music Maps and (b) to understand the human way of perception of music.

4.2 Expanding Collections

Since audio collections grow constantly, also Music Maps representing them must be constantly adopted. The main problem is that once a user is familiar with “his” Music Map it is very disturbing if the map changes dramatically which might happen when a Music Map is recreated.

As long as only few songs of a similar style already represented on the map are added there is no need to create a new map. Simply adding these songs to the Music Map is sufficient. However, if the range or the distribution of the different styles changes dramatically (e.g. by adding a new musical style) the map has to be retrained. But also in this case the Map should not change completely.

So main questions to address are:

1. At what point does a Music Map need to be recreated? How can this point be automatically determined?
2. How can the system ensure that the map does not change completely?
3. How can the changes on the map be appropriate displayed?

4.3 Path Merging

So far PocketSOM can act as remote control for the PlaySOM application. This is, however, limited to one single user. But when it comes to creating playlists for a group this concept does not reach far enough.

To allow multi-user playlist creation the approach is as follows: Multiple users send their trajectories or regions of interests on the map to the central server where these inputs will be further processed. The system tries to merge the received paths to on common playlists that fits all the user's requirements. There are several different different ways to combine paths and points sent by users:

- **Path Concatenation** With this most simple approach paths are concatenated one after the other. This might sound rather unsophisticated but it is, especially in combination with other techniques, challenging to find the best sequence of paths.
- **Path Clustering** With this approach two paths are taken and the average between is calculated and so snapped together. This technique has problems dealing with paths of different length. To avoid such problems paths might be first split into paths of a fixed length and after the clustering reconcatenated.
- **Point Clustering** After converting paths into a series of points these points are then clustered and from the centroids of these clusters a new path is calculated. The main questions for this approach is (a) how many points are used per path, (b) how many clusters are created, and (c) how do the resulting points create a new path?
- **Point Discretisation** Instead of converting paths to their points on the map the grid that lies behind the map is taken into account. Every unit on the grid that is covered by the path is marked. The more often a unit is marked the more weight it will gain in the following clustering process. Again, after calculating the clusters an new path based on the centroids is created. The questions (b) and (c) from the previous point also apply to this approach.

References

1. Teuvo Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, Heidelberg, 1995.
2. Thomas Lidy and Andreas Rauber. Evaluation of feature extractors and psycho-acoustic transformations for music genre classification. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 34–41, London, UK, September 11-15 2005.
3. Robert Neumayer, Michael Dittenbach, and Andreas Rauber. PlaySOM and PocketSOMPlayer – alternative interfaces to large music collections. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 618–623, London, UK, September 11-15 2005.
4. Robert Neumayer, Jakob Frank, Peter Hlavac, Thomas Lidy, and Andreas Rauber. Bringing mobile based map access to digital audio to the end user. In *Proceedings of the 14th International Conference on Image Analysis and Processing Workshops (ICIAP'07), 1st Workshop on Video and Multimedia Digital Libraries (VMDL'07)*, pages 9–14, Modena, Italy, September 10-13 2007. IEEE.