

SoundAnchoring: Content-based Exploration of Music Collections with Anchored Self-Organized Maps

Leandro Collares

Department of Computer Science
University of Victoria
leco@cs.uvic.ca

Tiago Fernandes Tavares

School of Electrical and Computer Engineering
University of Campinas
tavares@dca.fee.unicamp.br

Joseph Feliciano

Department of Computer Science
University of Victoria
noelf@uvic.ca

Shelley Gao

Department of Computer Science
University of Victoria
syugao@gmail.com

George Tzanetakis

Department of Computer Science
University of Victoria
gtzan@cs.uvic.ca

Amy Gooch

Department of Computer Science
University of Victoria
amy.a.gooch@gmail.com

ABSTRACT

We present a content-based music collection exploration tool based on a variation of the Self-Organizing Map (SOM) algorithm. The tool, named SoundAnchoring, displays the music collection on a 2D frame and allows users to explicitly choose the locations of some data points known as anchors. By establishing the anchors' locations, users determine where clusters containing acoustically similar pieces of music will be placed on the 2D frame. User evaluation showed that the cluster location control provided by the anchoring process improved the experience of building playlists and exploring the music collection.

1. INTRODUCTION

Commonly used interfaces for organizing music collections, such as iTunes and Microsoft Media Player, rely on long sortable lists of text and allow listeners to interact with music libraries using textual metadata (e.g., artist name, track name, album name, genre, etc.). Text-based interfaces excel when the user is looking for specific tracks. However, these interfaces are not suited for indirect queries, such as finding tracks that sound like a given track. Furthermore, text-based interfaces do not give users the ability to quickly summarize an unknown music collection.

Content-Based music collection Visualization Interfaces (CBVIs), such as *Islands of Music* [1], *MusicBox* [2] and *MusicGalaxy* [3], use Music Information Retrieval (MIR) techniques to group tracks from a collection according to their auditory similarity. In these interfaces, acoustically similar tracks are placed together in clusters, whereas dissimilar tracks are placed further apart. Consequently, CBVIs can reveal relationships between tracks that would be difficult to detect using text-based interfaces.

A number of CBVIs rely on the Self-Organizing Map (SOM) [4] to organize the tracks of the music collection ac-

cording to acoustic similarities. In the traditional SOM algorithm, however, users cannot determine the positions of clusters containing acoustically similar tracks on the music space. Additionally, the clusters' positions are randomized between different executions of the algorithm. We believe these characteristics can have a negative impact on the user experience.

In order to address the previously described issues, this paper presents *SoundAnchoring*, a CBVI that not only emphasizes meaningful relationships between tracks, but also allows users to determine the general placement of track clusters themselves. With SoundAnchoring, users can customize the layout of the music space by choosing the locations of a small number of tracks. These 'anchor' tracks and their respective positions determine the locations of clusters containing acoustically similar tracks on the music space. Such features allow users to create playlists easily without giving up control over which tracks are added. SoundAnchoring turns a music library into an interactive music space in three steps: feature extraction, organization and visualization.

Feature extraction involves calculating an n -dimensional 'feature' vector for each track. Since each element of the feature vector is an acoustic descriptor, tracks whose feature vectors are similar will be acoustically similar.

In the organization stage, we use AnchoredSOM, a variation of the traditional SOM algorithm. AnchoredSOM maps the music collection into a 2D representation that can be displayed on a screen. Moreover, AnchoredSOM gives users the power to determine the positions of clusters containing acoustically similar tracks on the 2D music space.

Lastly, the output of AnchoredSOM is used to render a visualization of the music collection. SoundAnchoring provides users with different ways to interact with the collection. If present, metadata is used to enrich the visualization. An outline of SoundAnchoring is depicted in Figure 1.

SoundAnchoring was evaluated through a user study. The anchoring process was evaluated positively. Ultimately, users felt that SoundAnchoring was easier to use than the control system, which was based on the traditional SOM algorithm. Thus, we conclude that the ability to choose

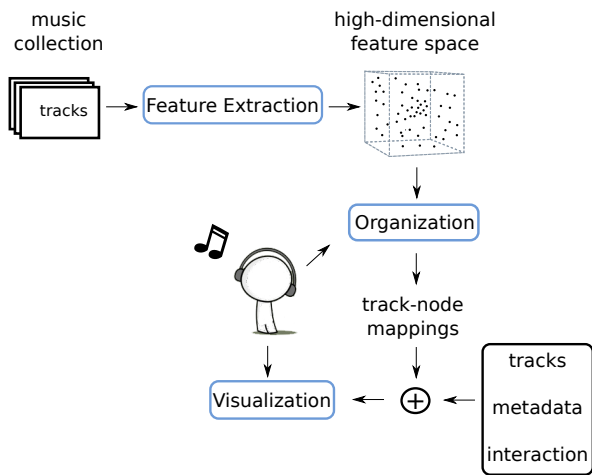


Figure 1: Outline of SoundAnchoring. A feature vector is computed for each track of the music collection. The set of feature vectors is a high-dimensional space that is mapped to two dimensions using the AnchoredSOM algorithm. The output of the algorithm is used to create a visualization of the music space. Users customize the positions of clusters containing acoustically similar tracks on the music space by choosing the locations of anchors.

anchors and their positions on the music space is an important feature in CBVIs that employ SOMs.

The remainder of the paper is organized as following:

- Section 2 contains related work on CBVIs that use SOMs.
- Section 3 describes the design of SoundAnchoring, with an emphasis on the organization and visualization stages.
- Section 4 describes the user study conducted to evaluate SoundAnchoring.
- Section 5 presents and discusses the results of the user study.
- Section 6 closes the paper with conclusive remarks and possible avenues of future work.

2. RELATED WORK

The SOM has been frequently employed in content-based interfaces to generate visualizations of music collections. Other dimensionality reduction techniques used for music collection organization include Principal Component Analysis (PCA) and Multidimensional Scaling (MDS), employed in *MusicBox* [2] and in *MusicGalaxy* [3], respectively.

In SoundAnchoring, SOM is employed to make optimum use of screen space on mobile devices. Tolos et al. [5] and Muelder et al. [6] showed that the music space produced by PCA presents problems regarding the distribution of tracks. Mörchen et al. [7] suggested that since the output of PCA and MDS are coordinates in a 2-dimensional

plane, it is hard to recognize groups of similar tracks, unless these groups are clearly separated. By choosing suitable parameters for the SOM algorithm, we believe that the music space can be displayed in an aesthetic way and occurrences of regions completely devoid of tracks can be minimized.

The first interface for music collection exploration that employed SOMs, SOMeJB, was an adaption of a digital library system. Interfaces that employ SOMs have evolved since then by incorporating more possibilities of interaction and customization, and auditory feedback.

SOMeJB (SOM-extended Jukebox), devised by Rauber and Frühwirth [8], introduced the use of SOMs for music collection exploration but still relied heavily on text to represent the music space. SOMeJB extended the functionalities of the SOMLib digital library system [9], which could organize a collection of text documents according to their content. SOMeJB was aimed to enable users to browse a music collection without a particular track in mind. The music library visualization generated by SOMeJB comprised a grid with track names grouped according to acoustic similarities between tracks. Even though SOMeJB represented a major departure from metadata-based organization, text was still the principal element of the interface.

In *Islands of Music*, a SOM-based interface developed by Pampalk et al. [1], the importance of text was diminished. The goal of *Islands of Music* was to support the exploration of unknown music collections using a geographic map metaphor. Clusters containing similar tracks were visualized as islands, while tracks that could not be mapped to any of the islands were placed on the sea. Connections between clusters were represented by narrow strips of land. Within an island, mountains and hills depicted sub-clusters. It was also possible to enrich the visualization by adding text summarizing the characteristics of the clusters.

Islands of Music inspired several content-based interfaces that, in addition to employing the geographic metaphor, refined the possibilities of interaction between users and music collections. *PlaySOM*, developed by Neumayer et al. [10], relied on the same metaphor of *Islands of Music*. *PlaySOM* improved the interaction with the music library by allowing users to add all tracks of a SOM node to a playlist.

Further refinements in interfaces using SOMs employed audio to assist in navigating music collections. *Sonic SOM*, devised by Lübbers [11], featured spatial music playback to provide users with an immersing experience. Knees et al. [12] developed *nepTune*, a 3D version of *Islands of Music* [1]. In *nepTune*, the user would navigate the music collection with a video game controller while tracks close to the listener's current position were played using a 5.1 surround system. Metadata retrieved from the Internet, such as tags and artist-related images, were displayed on screen to describe the track being played. Lübbers and Jarke [13] conceived an interface similar to *nepTune*. Valleys and hills replaced islands and oceans, respectively. Auditory feedback was enhanced by attenuating the volume of the tracks that deviated from the user's focus of attention.

A system developed by Brazil et al. [14, 15] combines both visual and auditory feedback for navigation. In this system, a user would navigate a sound space by means of a cursor surrounded by an ‘aura’. All sounds encompassed by the aura would be played simultaneously, but spatially arranged according to their distances from the cursor.

Although computer-based organization of music is an important tool for exploring of music collections, the perception of music is known to be highly subjective [16]. Thus, different listeners employ different methods to explore their music libraries. In order to accommodate these methods, interfaces should ideally adapt to the user’s behaviour.

The previously described work of Lübbers and Jarke [13] allowed users to customize the environment by changing the positions of the tracks, adding landmarks, or building and destroying hills. These actions would modify the similarity model employed to organize the music collection and thus cause the system to re-build the environment to reflect the user’s preferences.

A similar approach was adopted by Stober and Nürnberger [17], who developed *BeatlesExplorer*. In this interface, a music collection comprising 282 Beatles tracks was organized using SOMs. A user could drag and drop tracks between nodes, which would make the system relocate other tracks so that the collection organization could satisfy the user’s needs.

Interfaces for music collection exploration with smartphones and tablets in mind were also developed. Such interfaces benefited from the increase in processing power and storage for mobile devices and new possibilities of user interaction provided by touch-based screens. *PocketSOM-Player*, created by Neumayer et al. [10], was an interface derived from PlaySOM geared towards mobile devices. In *PocketSOMPlayer*, tracks could be added to a playlist by drawing trajectories on the music collection visualization.

Improvements in multi-touch gesture interaction stimulated the design of interfaces that allowed visually-impaired individuals to explore music collections without relying on the WIMP (window, icon, menu, pointer) paradigm. In the prototype developed by Tzanetakis et al. [18] for iOS devices, a random track would begin to play as soon as the user tapped on a square of the SOM grid. Moving one finger across squares would cause tracks from adjacent squares to cross-fade with each other, thereby generating auditory feedback.

With SoundAnchoring, users choose ‘anchor’ tracks and their positions on the music space. AnchoredSOM, a variation on the traditional SOM algorithm, places acoustically similar tracks on the neighbourhood of each anchor. Therefore, users are able to determine both the locations of clusters on the music space and their auditory content.

The concept of anchoring was introduced by Giorgetti et al. [19], who employed SOMs for localization in wireless sensor networks. The algorithm devised by Giorgetti et al. did not modify the weight vectors of nodes that contain anchors. Furthermore, Giorgetti et al.’s algorithm replaced the input vector with the node’s weight vector when the input vector was mapped to an anchor node. In Anchored-

SOM, weight vectors of all nodes are modified, while input vectors remain constant.

SoundAnchoring allows users to select tracks individually or by moving one finger over the music space, based on the implementation of Neumayer et al. [10]. While moving the finger on the device’s surface, users receive auditory feedback derived from the mechanism designed by Tzanetakis et al. [18] for assistive browsing.

3. SOUNDANCHORING DESIGN

The design of SoundAnchoring is comprised of three steps: feature extraction, organization and visualization. Feature extraction consists of representing each track of the collection as a vector of features that characterize the musical content. Tracks that sound alike are close to each other in the feature space. In organization, the high-dimensional feature space is reduced to a 2-dimensional representation. The topology of the feature space is preserved during this step. Finally, the output of the organization stage is used to produce a visualization of the music space. Users can interact with this customizable music space visualization and build playlists.

Feature extraction is carried out on a desktop computer, as it is independent from user interaction. Organization and visualization take place on an iPad 2. The forthcoming subsections present details pertaining to each step.

3.1 Feature Extraction

Feature extraction is the computation of a single feature vector for each track of the music collection. Before performing feature extraction, the first and the last fifteen seconds of each track are removed to avoid lead-in and lead-out effects. The audio clips are then divided into 23-ms frames, with a 12.5-ms overlap. Each frame is multiplied by a Hanning window and has its Discrete Fourier Transform (DFT) calculated. After that, we calculate a set of features for each frame. Later, the value series for each feature is divided into a 1-second frame, with length of 12.5 milliseconds between the beginning of each frame. The mean and variance of each frame are computed, generating two series f_μ and f_σ . Finally, the mean and variance of f_μ and f_σ are calculated. Therefore, there are four elements in the feature vector for each acoustic feature calculated.

The sixteen acoustic features employed in SoundAnchoring are frequently used in automatic genre classification tasks: thirteen MFCCs (Mel-Frequency Cepstral Coefficients), Spectral Centroid, Spectral Rolloff and Spectral Flux [20]. After feature extraction, each audio clip yields a 64-dimensional feature vector. Tracks that have similar feature vectors sound alike. AnchoredSOM reduces the 64-dimensional feature space to two dimensions for easy visualization. Acoustically similar tracks are placed close to each other on the 2D music space.

3.2 Organization

The organization stage maps the 64-dimensional feature space to discrete coordinates on a grid using SOM. This dimensionality reduction technique preserves the topology

of the high-dimensional space as much as possible; tracks that have similar feature vectors should be placed close to each other, whereas tracks that have dissimilar feature vectors should be apart in the 2-dimensional space. SoundAnchoring employs AnchoredSOM to allow the user to define the location of some specific tracks or anchors.

The traditional SOM is an artificial neural network in which nodes are arranged in a 2-dimensional rectangular grid. During the execution of the SOM algorithm, the neural network is iteratively trained with input vectors, namely the feature vectors computed during feature extraction. At the end of the execution, different parts of the network are optimized to respond to certain input patterns.

Each node of the SOM is characterized by two parameters: a position in the two-dimensional space and a weight vector of the same dimensionality as the feature vectors: 64. When a feature vector is presented to the network, the best matching node (BMN), i.e., the node whose weight vector is the most similar to the feature vector is determined. The feature vector, which corresponds to one track of the music collection, is mapped to the BMN. The BMN's weight vector is updated to resemble the feature vector. Weight vectors of the BMN's neighbouring nodes are also updated towards the feature vector. The magnitude of the change in the neighbouring nodes' weight vectors, which is determined by the learning rate, decreases with time and distance. The neighbourhood size also decreases with time. After several iterations, different parts of the network will have similar weight vectors and, consequently, will respond similarly to certain feature vectors.

In visualizations of music collections based on the traditional SOM algorithm, tracks that sound similar tend to be close to each other. The SOM algorithm, however, does not have information regarding genre labels as only feature vectors are used as input to the algorithm. Thus, the locations of genre clusters are an emergent property of the SOM.

The weight vectors are usually initialized with small random values. Consequently, the positions of clusters containing acoustically similar tracks on the music space cannot be determined in advance by the user. Moreover, the position of a given cluster containing similar tracks is likely to vary between executions of the traditional SOM algorithm, as shown in Figures 2a-2d. We believe this scenario has a negative impact on the user experience. In order to alleviate the situation, we introduce AnchoredSOM, a variation on the traditional SOM algorithm.

3.2.1 AnchoredSOM

AnchoredSOM allows users to choose the locations of 'anchor' data points on the SOM, which correspond to tracks in the music collection. The anchors will attract similar tracks to their neighbourhoods. AnchoredSOM consists of four stages, detailed below:

- **Stage 0.** This stage is analogous to the initialization of the traditional SOM. In AnchoredSOM, however, node weight vectors are initialized with feature vectors randomly chosen from the high dimensional feature space. This approach speeds up the convergence

of the SOM algorithm.

- **Stage 1.** In this stage, only feature vectors of the anchors are presented to the SOM for i_1 iterations. Both the initial learning rate, L_0 , and the initial neighbourhood size, σ_0 , have high values to cause significant changes to the weight vectors of the entire SOM.
- **Stage 2.** Only feature vectors of the anchors are presented to the SOM for i_2 iterations. In stage 2, however, the initial learning rate, L_0 , and the initial neighbourhood size, σ_0 , are low to bring small changes to localized areas of the SOM.
- **Stage 3.** For each of the i_3 iterations, the input of the entire feature set to the SOM is followed by m occasions on which only the anchors' feature vectors are presented to the SOM. The input of anchors' feature vectors for m successive times within one iteration keeps the weight vectors of nodes surrounding the anchors' nodes similar to the anchors' feature vectors.

In our implementation, we employed the Euclidean distance for measuring the similarity between feature vectors. Learning and neighbourhood functions are exponentially-decaying with time. The values for the number of iterations, initial learning rate and initial neighbourhood size were empirically determined. The size of the grid is based on the number of tracks in the music collection.

Figures 2e-2h show that AnchoredSOM lends itself to setting the positions of clusters containing similar music. AnchoredSOM performs better with genres that are distinct and well-localized, such as the classical genre. With acoustically diverse genres, such as the pop genre, the tracks will be more loosely dispersed on the grid.

3.2.2 Number of Anchors

A pilot study was conducted to determine the number of anchors that would be used in SoundAnchoring. Participants were told that we had designed an interface able to organize their entire music collection on a 2D grid in a logical manner. They were also told that information was being collected regarding the number of music genres people needed to organize their collections. Participants received a sheet of paper containing a 10x10 grid and a table to make colour-genre associations.

Firstly, individuals had to complete the table with the minimum set of genres they deemed necessary to categorize their collection effectively. Some major categories were presented but they were encouraged to add more genres if any genres were unrepresented. After picking the genres, participants were asked to colour the squares next to the genres using a set of crayons. Later, participants were asked to choose one square of the grid to act as the centre point of each genre. Similar tracks would be grouped around that square. Glass tokens were provided to help participants space out the chosen squares before colouring them. Most participants chose five categories and thus SoundAnchoring uses five anchors of different genres.

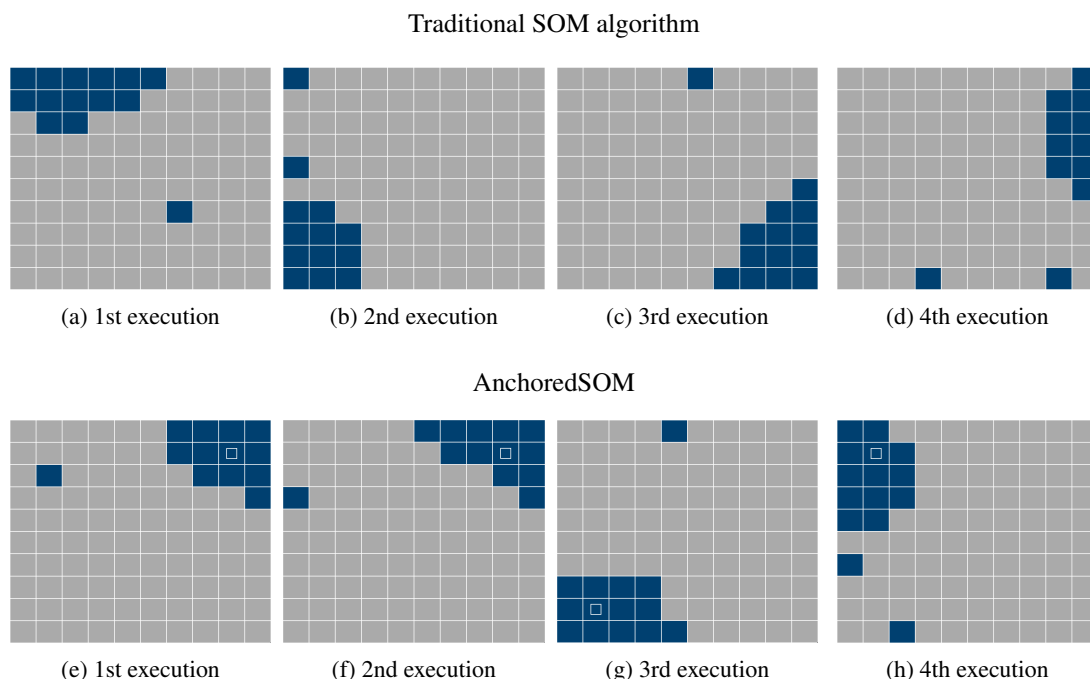


Figure 2: Topological mapping of clusters containing classical tracks, in blue. Traditional SOM, subfigures a-d: the location of the classical cluster varies drastically with each execution of the algorithm. AnchoredSOM, subfigures e-h: the same white-marked anchor track was used to maintain the position of the classical cluster in (e, f). When the same anchor track is placed on a different node, the other classical tracks remained clustered around it (g, h).

3.3 Visualization

The output of AnchoredSOM is employed to generate a visualization of the music collection. In our implementation, interactions with the music collection are based on the Apple Cocoa Touch API (Application Programming Interface). In order to get to the final screen, which contains the music space, users go through a sequence of screens and make choices that influence the organization and the appearance of the music space. The sequence of screens aims to lower the cognitive load on the user.

In SoundAnchoring, colours convey information on genres. As user studies have shown no basis for universal genre-colour mappings [21], SoundAnchoring allows users to make genre-colour associations using seven palettes, derived from Eisemann’s work [22]. Eisemann built associations between colours and abstract categories such as ‘capricious’, ‘classic’, ‘earthy’, ‘playful’, ‘spicy’, ‘warm’, etc. The aforementioned categories referred to moods that each colour grouping evoked when utilized in advertisements, product packaging and print layouts. The colours of each grouping created by Eisemann were chosen from the Pantone Matching System, a de facto colour space standard in publishing, fabric and plastics. These predefined colour palettes give users some freedom to assign colours to genres and have a positive bearing on the aesthetics of the music space visualizations.

Classifying music by genre is challenging, as there is often overlapping between genres and disagreement on the label set used for classification [23]. Genres, however, are usually employed to narrow down the number of choices when browsing music for entertainment reasons [24]. There-

fore, genres provide users with a familiar vantage point to start exploring their music collections.

After selecting a colour palette and building genre-colour associations, users choose five anchors from the music collection and place them on the grid. The anchors’ feature vectors and locations are presented to AnchoredSOM, along with the feature vectors of the other tracks of the music collection. AnchoredSOM then maps the tracks to nodes of the SOM.

3.3.1 Interaction with Music Collection

The SoundAnchoring interface (Figure 3) displays the entire music collection on a grid. Users interact with the music collection using different gestures.

By tapping on one of the nodes of the grid, users will see a list of tracks mapped to that node by AnchoredSOM. Single-tapping on the track gives audio feedback. Double-tapping on the track adds it to the playlist. This action is similar to building a playlist by selecting tracks individually in text-based interfaces. With the SOM, however, acoustically similar tracks will be either in the same node or in neighbouring ones.

Instead of listing the tracks of a certain node and adding tracks to the playlist individually, users can alternatively moving one finger over the grid to add multiple tracks to the playlist. As the user performs this gesture, known as ‘sketching’, SoundAnchoring randomly adds one track of each node activated by the user’s finger to the playlist. The user also receives aural and visual feedback while sketching. Excerpts of the randomly chosen tracks cross-fade with each other as the user moves the finger across nodes as a way of providing auditory feedback to users. The opac-

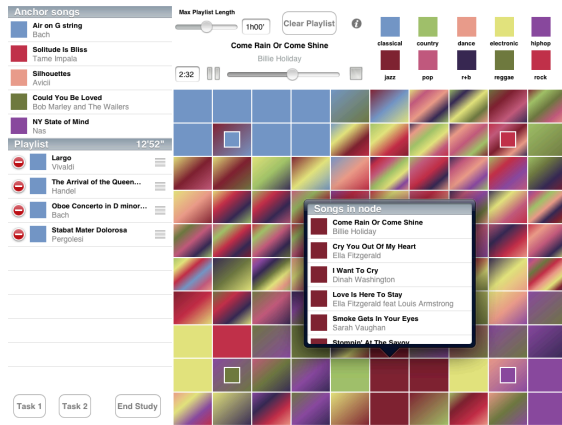


Figure 3: SoundAnchoring interface. Tapping on a node reveals tracks that have been mapped to that node. Genre buttons allow users to limit the number of genres displayed on the music space. Playlists can be built by selecting tracks individually or ‘sketching’ on the surface, which causes SoundAnchoring to randomly choose one track from each node.

ity of the nodes that have been activated oscillates for a few seconds giving the impression of a trail on the grid.

Finally, genre masks refine the use of genres as a familiar vantage point to explore music libraries. Genre buttons coloured according to the genre-colour associations previously made are employed to filter genres that are displayed. If a genre is filtered out, both the colour assigned to that genre and the tracks belonging to it disappear from the grid. Consequently, these tracks are not listed when the user taps on a node. Furthermore, sketching across nodes does not add tracks from the filtered-out genre to the playlist. Therefore, genre masks give users more flexibility to explore the music space.

4. EVALUATION

For evaluation we conducted a user study in which each one of the twenty-one participants (eleven females and ten males) performed tasks in two systems with the same visual interface: SoundAnchoring (SA), which allows individuals to determine the position of anchors on the music space, and a Control System (CS), which loads pre-calculated maps generated using the traditional SOM algorithm.

The study took place in a prepared office room. SoundAnchoring and the Control System were loaded in two iPads 2. Participants were randomly assigned to start working with either SA or CS to compensate for learning effects.

Subjects performed two tasks, with no enforced time limits. Task 1 was conceived to raise awareness for the mapping of similar tracks to the same node or neighbouring nodes of the SOM. Participants were required to tap on one square of the grid and listen to the tracks of that square, then its adjacent squares. These steps were repeated with two other squares, distant from the first square and from each other. Task 2 was the creation of a playlist. Slips of

paper containing descriptions of different scenarios were placed face down. Participants were asked to pick one slip of paper and build a playlist of at least thirty minutes containing a minimum of three genres that would match the scenario described.

After using each system, subjects rated a set of eighteen statements using a 6-point scale (from zero to five). Subjects were also encouraged to write about positive and negative aspects of each system, as well as recommendations for improvement.

5. RESULTS AND DISCUSSION

The mean values for each statement were calculated and the statistical significance of the differences between systems were computed using Fisher’s randomization test [25]. The statements, mean values and p -values are shown in Table 1.

In most statements, the mean rate difference is not statistically significant ($p > 0.05$). A remarkable exception is statement 10 (“Getting the system to do what I wanted was easy”), which shows that SoundAnchoring is consistently evaluated as easier to use than the Control System. However, most of the results are inconclusive, which necessitates a qualitative analysis of the textual feedback provided by the subjects.

Overall, both SA and CS were favourable reviewed by participants as shown by mean rates for statements 4-6, 9, 12, 15 and 18 (Table 1). Words employed to describe both SOM-based systems: “intuitive”, “easy to use”, “aesthetically appealing”, “interesting”, “flexible”, “user-friendly”, and “entertaining”. More elaborate comments on the interface included: “easy to sample-listen to songs”, “a fun way to browse a music collection”, “good for exploring unfamiliar music collections”, “easy to find songs similar to known ones you like”, “similar songs are actually similar”, “does a good job of grouping similar music”, “great to access songs you have forgotten about” and “nice mapping from sounds to graphics”.

Comments suggest that participants perceived the visualization of the music collection using SOMs and the grouping of acoustically similar tracks as positive. Therefore, the clustering process was able to retrieve useful information from the music collection and display it properly. Moreover, the feedback shows that content-based music collection visualization is an efficient approach to music collection exploration.

Playlist creation was mentioned in comments such as “It is easy to build accurate playlists for specific scenarios”, “Making a playlist becomes fun instead of a chore” and “easy to take playlist in a new sound direction that suits your inspiration”. By analyzing user-system interactions that were logged during the user study, we realized that most participants added tracks to the playlist by tapping on each node and selecting tracks individually. This behaviour was reflected in comments such as “It can be time-consuming to make a playlist”, “I wanted to have total control over the songs added to the playlist, so I had to tap on all the grid boxes to get to know the songs”. One participant particularly liked the sketching gesture for creating

Statement	Mean rate		<i>p</i> -value
	SA	CS	
1. Please rate the playlist you created in task 2.	4.2	4.1	0.83
2. The interactions with the interface were natural.	3.8	3.7	1.0
3. I was unable to anticipate what would happen next in response to the actions I performed.	1.2	1.4	0.67
4. The amount of controls available to perform the tasks was adequate.	4.0	4.2	0.22
5. The auditory aspects of the interface appealed to me.	4.3	4.2	0.74
6. The visual aspects of the interface were unappealing to me.	0.9	1.0	0.72
7. It was impossible to get involved in the experiment to the extent of losing track of time.	1.2	1.6	0.39
8. I felt proficient in interacting with the interface at the end of the experiment.	3.6	3.4	0.64
9. The interface was unresponsive to actions I initiated (or performed).	0.8	0.6	0.58
10. Getting the system to do what I wanted was easy.	4.3	3.8	0.03
11. I would consider replacing my current application for music exploration with one based on the system tested.	2.6	3.2	0.07
12. Learning how to use the system was difficult.	0.8	1.0	0.70
13. I disliked creating playlists with the system.	1.0	1.0	1.0
14. The system is unsuitable for managing and exploring my music collection.	1.7	1.4	0.46
15. I enjoyed exploring the music collection with the system.	4.2	4.4	0.67
16. I can create playlists quickly by using the system.	2.9	3.1	0.54
17. I disliked the playlists created by using the system.	0.8	0.8	1.0
18. Please provide an overall rate for the system.	4.0	4.1	0.52

Table 1: Statements' mean rates for SoundAnchoring (SA) and the Control System (CS), and *p*-values. Better rates for each statement and the statistically significant *p*-value are shown in bold.

playlists: "Adding songs to the playlist by dragging my finger on the surface and listening to audio was a really nice feature I was impressed with". A slightly different opinion was expressed by another participant: "I really liked to be able to explore the collection sliding my finger on the surface but I think it shouldn't add the songs to the playlist when I do that. I can add the songs individually later". Even though there is some disagreement with regard to interaction, playlist creation using the interface was seen as enjoyable. Feedback from participants is supported by the mean rates for statements 1, 13 and 17 in Table 1. Therefore, the goal of building an interface in which building playlists would be engaging was achieved.

As for the anchoring mechanism, opinions were in general positive. Most participants stated it was useful: "With anchor songs I knew where to start browsing my music collection", "Close songs were actually similar to each other in the version with anchor songs", "I did like knowing where my anchor songs were as it was easier to figure out which types of songs were in the various areas of the grid", "Anchor songs helped me decide where to look for songs suitable to the situation given", "I would be interested in using a conventional system (album, artist, title) to explore my music collection and then selecting the anchors to browse similar songs". Only one participant claimed that "anchoring didn't help much". These statements show that anchors helped participants navigating the music collection. Moreover, subjects were able to adapt the music collection organization to their individual preferences by setting the clusters' positions on the grid. Such conclusions are in agreement with mean rates for statement 10.

Participants also provided invaluable suggestions to fur-

ther improve the user experience provided by SoundAnchoring. Among these suggestions are a zooming function to explore more thoroughly areas of the music space and a search function to locate specific tracks on the grid. Subjects would also like to add all the tracks of a node to the playlist with only one gesture. With regard to anchoring, participants would like the interface to recommend anchors based on listening habits. Therefore, SoundAnchoring should incorporate more possibilities of interaction to cater for different ways of exploring music collections, and learn from users' behaviour.

6. CONCLUSION

This paper presents SoundAnchoring, a content-based music visualization interface that maps the music library to a 2D space. With SoundAnchoring, users play an active role in the organization of the music space by choosing where clusters containing acoustically similar tracks will be located.

A user study was carried out to evaluate SoundAnchoring. The ability to modify the topology of the music visualization, along with gestural control and other interface-related features, delivered a positive user experience with regard to playlist creation. Despite encouraging results, SoundAnchoring can be improved in several ways. Immediate enhancements comprise the addition of new gestures suggested by user study participants.

As for future work, we intend to perform an objective evaluation of AnchoredSOM that takes different feature sets and algorithm parameters into consideration. A long-term user study involving a larger number of participants

could more comprehensively evaluate the real-world applicability of SoundAnchoring. Further research avenues include the use of graphics processing units (GPUs) and cloud computing to improve the performance of the feature extraction and organization stages.

7. REFERENCES

- [1] E. Pampalk, A. Rauber, and D. Merkl, "Content-based organization and visualization of music archives," in *Proceedings of the 10th ACM International Conference on Multimedia*. ACM, 2002, pp. 570–579.
- [2] A. S. Lillie, "MusicBox: Navigating the space of your music," Master's thesis, Massachusetts Institute of Technology, 2008.
- [3] S. Stober and A. Nürnberger, "Musicgalaxy - an adaptive user-interface for exploratory music retrieval," in *Proc of 7th Sound and Music Computing Conference*, 2010.
- [4] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.
- [5] M. Tolos, R. Tato, and T. Kemp, "Mood-based navigation through large collections of musical data," in *Consumer Communications and Networking Conference, 2005. CCNC. 2005 Second IEEE*. IEEE, 2005, pp. 71–75.
- [6] C. Muelder, T. Provan, and K.-L. Ma, "Content based graph visualization of audio data for music library navigation," in *IEEE International Symposium on Multimedia (ISM)*. IEEE, 2010, pp. 129–136.
- [7] F. Mörchén, A. Ultsch, M. Nöcker, and C. Stamm, "Visual mining in music collections," *From Data and Information Analysis to Knowledge Engineering*, pp. 724–731, 2006.
- [8] A. Rauber and M. Frühwirth, "Automatically analyzing and organizing music archives," *Research and Advanced Technology for Digital Libraries*, pp. 402–414, 2001.
- [9] A. Rauber and D. Merkl, "The SOMlib digital library system," *Research and Advanced Technology for Digital Libraries*, pp. 852–852, 1999.
- [10] R. Neumayer, M. Dittenbach, and A. Rauber, "PlaySOM and pocketSOMplayer, alternative interfaces to large music collections," in *Proc. of ISMIR*, vol. 5, 2005.
- [11] D. Lübbers, "SoniXplorer: Combining visualization and auralization for content-based exploration of music collections," in *Proc. of ISMIR*, 2005, pp. 590–593.
- [12] P. Knees, M. Schedl, T. Pohle, and G. Widmer, "An innovative three-dimensional user interface for exploring music collections enriched with meta-information from the web," in *Proceedings of the ACM Multimedia*, 2006, pp. 17–24.
- [13] D. Lübbers and M. Jarke, "Adaptive multimodal exploration of music collections," in *Proc. of ISMIR*, vol. 2009, 2009.
- [14] E. Brazil, M. Fernström, G. Tzanetakis, and P. Cook, "Enhancing sonic browsing using audio information retrieval," in *International Conference on Auditory Display ICAD-02, Kyoto, Japan*, 2002.
- [15] E. Brazil and M. Fernström, "Audio information browsing with the sonic browser," in *Coordinated and Multiple Views in Exploratory Visualization, 2003. Proceedings. International Conference on*. IEEE, 2003, pp. 26–31.
- [16] J. S. Downie, "Music information retrieval," *Annual review of information science and technology*, vol. 37, no. 1, pp. 295–340, 2003.
- [17] S. Stober and A. Nürnberger, "Towards user-adaptive structuring and organization of music collections," *Adaptive Multimedia Retrieval. Identifying, Summarizing, and Recommending Image and Music*, pp. 53–65, 2010.
- [18] G. Tzanetakis, M. S. Benning, S. R. Ness, D. Minifie, and N. Livingston, "Assistive music browsing using self-organizing maps," in *Proceedings of the 2nd International Conference on Pervasive Technologies Related to Assistive Environments*. ACM, 2009, pp. 3:1–3:7.
- [19] G. Giorgetti, S. Gupta, and G. Manes, "Wireless localization using self-organizing maps," in *Proceedings of the 6th international conference on Information processing in sensor networks*. ACM, 2007, pp. 293–302.
- [20] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.
- [21] J. Holm, A. Aaltonen, and H. Siirtola, "Associating colours with musical genres," *Journal of New Music Research*, vol. 38, no. 1, pp. 87–100, 2009.
- [22] L. Eisemann, *Pantone's Guide to Communicating with Color*. Grafton Press, Ltd., Florida, 2000.
- [23] M. Sordo, Ò. Celma, M. Blech, and E. Gaus, "The quest for musical genres: Do the experts and the wisdom of crowds agree?" in *Proceedings of the 9th International Conference on Music Information Retrieval*, 2008, pp. 255–260.
- [24] A. Laplante, "Users relevance criteria in music retrieval in everyday life: an exploratory study," in *Proceedings of the 11th International Society for Music Information Retrieval Conference*, 2010, pp. 601–606.
- [25] R. A. Fisher, *The Design of Experiments*. Hafner Publishing Company, New York, 1935.