SoundAnchoring: Personalizing music spaces with anchors

by

Leandro Collares de Oliveira
B.Sc., Universidade Federal de Minas Gerais, 1998

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

SoundAnchoring: Personalizing music spaces with anchors

by

Leandro Collares de Oliveira
B.Sc., Universidade Federal de Minas Gerais, 1998

Supervisory Committee

Dr. Yvonne Coady, Co-Supervisor
(Department of Computer Science)

Dr. Amy Gooch, Co-Supervisor
(Department of Computer Science)

Dr. George Tzanetakis, Departmental Member
(Department of Computer Science)

Supervisory Committee

Dr. Yvonne Coady, Co-Supervisor
(Department of Computer Science)

Dr. Amy Gooch, Co-Supervisor
(Department of Computer Science)

Dr. George Tzanetakis, Departmental Member
(Department of Computer Science)

ABSTRACT

Several content-based interfaces for music collection exploration rely on Self-Organizing Maps (SOMs) to produce 2D or 3D visualizations of music spaces. In these visualizations, perceptually similar songs are clustered together. The positions of clusters containing similar songs, however, cannot be determined in advance due to particularities of the traditional SOM algorithm. In this thesis, I propose a variation on the traditional algorithm named anchoredSOM. This variation avoids changes in the positions of the aforementioned clusters. Moreover, anchoredSOM allows users to personalize the music space by choosing the locations of clusters containing perceptually similar tracks. This thesis introduces SoundAnchoring, an interface for music collection exploration featuring anchoredSOM. SoundAnchoring is evaluated by means of a user study. Results show that SoundAnchoring offers engaging ways to explore music collections and build playlists.

Contents

List of Tables

List of Figures

Acknowledgements

Firstly, I would like to thank Yvonne Coady, Amy Gooch, and George Tzanetakis for the guidance and encouragement they gave me throughout the graduate program. I would like to extend my gratitude to Ulrike Stege for keeping me on the right path. Appreciation is also extended to Wendy Beggs, Esther Lee, Nancy Chan, and Erin Robinson for their patience and willingness to help me with all administrative issues. I would also like to thank Bill Gorman, Bette Bultena, and Victoria Li for the continuous support at the consultants' office. I acknowledge Murilo Gomes, Renato Mesquita, Eduardo Winter, Leonardo Torres, and Leonardo Pimenta, who made the dream of doing graduate studies in Canada possible.

I would like to acknowledge undergraduate and graduate students in the graphics, visualization, modsquad, and music labs for their sound advice on different aspects of this project. I would like to thank Miki Sangawa, Noel Feliciano, Rob Kelly, Brendan Clement, Dandan Huang, Shelley Gao, Jeremy Long, Steven Ness, Gabrielle Odowichuk, Chris Matthews, Jen Baldwin and Dean Pucsek for their invaluable help along the way. I also thank the participants of the user study for donating their time and providing important feedback and suggestions.

I would like to acknowledge Adriano, Polly, John, and Bente for inspiring me with perseverance and kindness. I thank Josie, Thais, Flávia, Tiago, and Dee for their love and help. I also extend my gratitude to Erik, Luiza, Joyce, Bruno, Bruce, Rafael, Fabiane, Luciana, and Chris for always providing me with a welcoming environment in Canada. My heartfelt appreciation is extended to all friends who jazz up my life every day. Last but not least, I would like to thank my parents, my sister, Mel, and other family members for their unconditional love and support.

> *To live only for some future goal is shallow. It's the sides of the mountains that sustain life, not the top. Here's where things grow.*
>
> Robert M. Pirsig

Dedicated to my parents, my sister, and Mel.

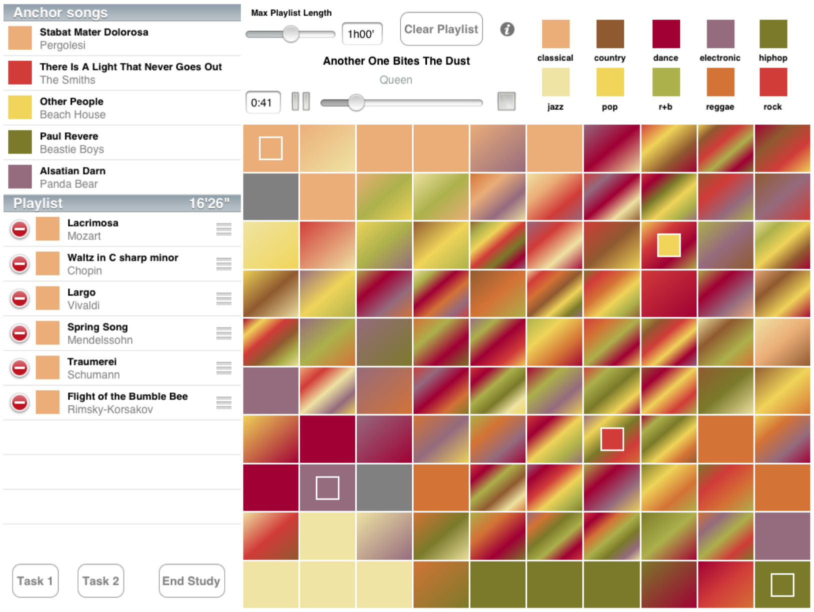
Chapter 1

Introduction

Several events in diverse technological fields have made possible to store thousands of songs on digital devices. The increasing size of music collections poses challenges with regard to organization and browsing. Text-based interfaces, such as iTunes or Microsoft Media Player, allow users to find in their collections specific songs they want to listen to, e.g., "Something good can work" by Two Door Cinema Club. These interfaces, however, cannot help users if they do not know what they want to listen to or if they know it but can only explain it vaguely.

Suppose an individual is riding the bus to school to take a final exam and would like to build a suitable playlist comprising songs that are soothing, yet energetic and fun. In this scenario, building a playlist using a text-based interface would mean navigating lists of text to select adequate songs individually. With content-based interfaces, however, users can interact with their music collections without relying exclusively on text. These interfaces organize the music collection spatially according to the similarity of the tracks, i.e., songs that sound similar will be close, whereas dissimilar songs will be distant from each other on the screen. Therefore, content-based interfaces allow users to explore their music collections serendipitously, find music that fits a given scenario and even unveil underlying affinities between pieces of music.

This thesis introduces *SoundAnchoring*, depicted in Figure 1.1. SoundAnchoring is a content-based interface featuring a novel algorithm termed anchoredSOM. This algorithm attempts to address a limitation of the traditional Self-Organizing Map algorithm. Furthermore, this research presents the results of a user study carried out to assess SoundAnchoring.



Anchor songs

Stabat Mater Dolorosa
Pergolesi

There Is A Light That Never Goes Out
The Smiths

Other People
Beach House

Paul Revere
Beastie Boys

Alsatian Darn
Panda Bear

Playlist 16'26"

Lacrimosa
Mozart

Waltz in C sharp minor
Chopin

Largo
Vivaldi

Spring Song
Mendelssohn

Traumerei
Schumann

Flight of the Bumble Bee
Rimsky-Korsakov

Another One Bites The Dust
Queen

Figure 1.1: SoundAnchoring, content-based interface for music collection exploration introduced and evaluated in this thesis.

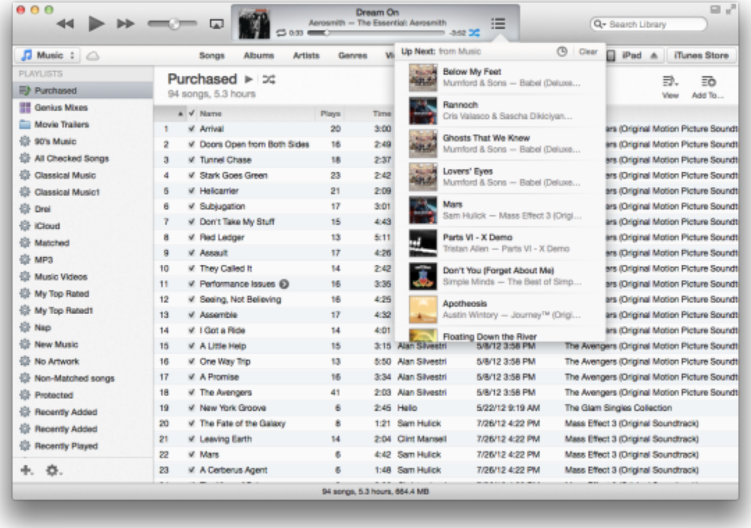
1.1 Motivation and contributions

Developments in different fields such as audio compression, networks, and digital storage have changed the way listeners consume music and created challenges regarding the organization of music collections. Popular applications used for organizing music collections might not be able to fully address these challenges.

Research on audio compression yielded the MP3 (MPEG-1 Audio Layer III) format that can deliver high quality audio with reduced storage footprint [45]. The advent of Napster, Kazaa and other peer-to-peer services as of 1999 and the increasing penetration of broadband made file sharing widespread [49,67]. Online stores like iTunes and Amazon MP3 offer millions of songs to listeners. Storage prices have been decreasing for hard disks as well as for flash memory [31, 63]. In this context, it is fairly easy to build personal libraries comprising thousands of songs. Organizing and accessing music collections, however, can be challenging.

Ideally users interact with music collections via direct and indirect queries. When users formulate a **direct query**, they know exactly what they want to listen to, e.g., “Air on G String” by Johann Sebastian Bach. Users, however, might not be able to pose a direct query, either because they do not have a specific song in mind or because they just want to get acquainted with a music collection. This scenario corresponds to an **indirect query** (or browsing), in which a user explores the collection with a certain degree of happenstance [55, 61].

As seen in Figure 1.2, common applications for music collection exploration, such as iTunes and Windows Media Player, are heavily based on **metadata** (or **contextual descriptors**) that describes the contents of the music collection. Metadata can be factual or cultural. While **factual metadata** contains objective information on a track, such as artist name, album name, duration and release year, **cultural metadata** presents subjective concepts, i.e., mood, emotion, genre [10]. Text-based interfaces typically allow users to sort the music collection using metadata, search for tracks using direct queries, shuffle and create playlists. This approach to music exploration is certainly familiar to most individuals that use personal computers since it is derived from the traditional filesystem metaphor, in which files are placed into folders.



Purchased

94 songs, 5.3 hours

	✓	Name	Plays	Time			
1	✓	Arrival	20	3:00			ers (Original Motion Picture Soundt
2	✓	Doors Open from Both Sides	16	2:49			ers (Original Motion Picture Soundt
3	✓	Tunnel Chase	18	2:37			ers (Original Motion Picture Soundt
4	✓	Stark Goes Green	23	2:42			ers (Original Motion Picture Soundt
5	✓	Helicarrier	21	2:09			ers (Original Motion Picture Soundt
6	✓	Subjugation	17	3:01			ers (Original Motion Picture Soundt
7	✓	Don't Take My Stuff	15	4:43			ers (Original Motion Picture Soundt
8	✓	Red Ledger	13	5:11			ers (Original Motion Picture Soundt
9	✓	Assault	17	4:26			ers (Original Motion Picture Soundt
10	✓	They Called It	14	2:42			ers (Original Motion Picture Soundt
11	✓	Performance Issues	16	3:36			ers (Original Motion Picture Soundt
12	✓	Seeing, Not Believing	16	4:25			ers (Original Motion Picture Soundt
13	✓	Assemble	17	4:32			ers (Original Motion Picture Soundt
14	✓	I Got a Ride	14	4:01			ers (Original Motion Picture Soundt
15	✓	A Little Help	15	3:15	Alan Silvestri	5/8/12 3:58 PM	The Avengers (Original Motion Picture Soundt
16	✓	One Way Trip	13	5:50	Alan Silvestri	5/8/12 3:58 PM	The Avengers (Original Motion Picture Soundt
17	✓	A Promise	16	3:34	Alan Silvestri	5/8/12 3:58 PM	The Avengers (Original Motion Picture Soundt
18	✓	The Avengers	41	2:03	Alan Silvestri	5/8/12 3:58 PM	The Avengers (Original Motion Picture Soundt
19	✓	New York Groove	6	2:45	Hello	5/22/12 9:19 AM	The Glam Singles Collection
20	✓	The Fate of the Galaxy	8	1:21	Sam Hulick	7/26/12 4:22 PM	Mass Effect 3 (Original Soundtrack)
21	✓	Leaving Earth	14	2:04	Clint Mansell	7/26/12 4:22 PM	Mass Effect 3 (Original Soundtrack)
22	✓	Mars	6	4:42	Sam Hulick	7/26/12 4:22 PM	Mass Effect 3 (Original Soundtrack)
23	✓	A Cerberus Agent	6	1:48	Sam Hulick	7/26/12 4:22 PM	Mass Effect 3 (Original Soundtrack)

Below My Feet
Mumford & Sons — Babel (Deluxe...
Rannoch
Cris Valasco & Sascha Dikiciyan...
Ghosts That We Knew
Mumford & Sons — Babel (Deluxe...
Lovers' Eyes
Mumford & Sons — Babel (Deluxe...
Mars
Sam Hulick — Mass Effect 3 (Origi...
Parts VI - X Demo
Tristan Allen — Parts VI - X Demo
Don't You (Forget About Me)
Simple Minds — The Best of Simp...
Apotheosis
Austin Wintory — Journey™ (Origi...
Floating Down the River

94 songs, 5.3 hours, 664.4 MB

Figure 1.2: iTunes, a popular text-based application for music collection exploration. Text-based applications perform well when users know exactly what they want to listen to, but do not allow users to explore the music collection serendipitiously.

While text-based interfaces perform well with regard to direct queries, they offer little support for indirect queries or browsing. Indeed, a study involving 5,000 iPod users revealed that 23% of songs of the music library were played 80% of the time, whereas 64% of the songs were never played [43]. Building a playlist comprising suitable songs for a certain occasion, e.g., commuting, exercising, working or studying, using a text-based interface would mean browsing through long sortable lists of text.

This task can be both time-consuming and tedious. The constant need to update playlists aggravates the previously mentioned situation. Lastly, interfaces based on text do not allow users to form a general impression of the collection without a thorough exploration.

The field of Music Information Retrieval (MIR) develops strategies for accessing music collections that meet the expectations of search and browse functionalities [10]. This field comprises computer science, information retrieval, musicology, music theory, audio engineering, digital signal processing, cognitive sciences, library science, publishing and law [22]. The interdisciplinary nature of MIR hinges on music's subjectivity. That is, each individual experiences music in a unique way, which depends on cultural background, knowledge, mood, etc. Downie [14] states "music ultimately exists in the mind of its perceiver. Therefore, the perception, appreciation, experience of music will vary not only across the multitudes of minds that apprehend it, but will also vary within each mind as the individual's mood, situation, and circumstances change".

One of the focus of MIR is the design of content-based interfaces to visualize and browse song collections. These interfaces employ **content-based descriptors** which are extracted from songs and convey information on "what the music sounds like" [33]. Although MIR "strives to develop novel interfaces in an effort to make the world's vast store of music accessible to all" [15], user studies to evaluate such interfaces are still few and far between [25, 65]. Without user studies, the evaluation of the real-world applicability of MIR systems is merely speculative [8].

This thesis makes two major contributions to the field of MIR. The first contribution is the design of a content-based interface termed *SoundAnchoring*. SoundAnchoring features an improvement on an existing technique for organizing music collections. The second contribution is a user study to evaluate SoundAnchoring. The next section presents detailed information on content-based interfaces for music collection exploration.

1.2 Content-based interfaces

As seen in Section 1.1, popular music collection exploration applications are based on metadata. Text-based interfaces for music collection exploration help users when they know which songs they want to listen to. These interfaces, however, do not aid users when they do not know what they are looking for or do not have the words to describe what they are looking for. In order to address this “semantic gap”, content-based interfaces have been developed by MIR researchers.

Content-based interfaces allow users to browse a large music collection effectively without a target song in mind. Tracks that are not listened to due to limitations imposed by text-based organization can be rediscovered. Moreover, content-based interfaces provide an overview of music collections without deep exploration. Generating playlists for specific scenarios is easier because similar songs are grouped together. Content-based interfaces, however, often offer weak support for direct queries. In order to address this shortcoming, content-based interfaces are usually enriched by metadata.

The following stages are required to design a content-based interface for exploration of music collections: Feature Extraction, Organization and Visualization. Figure 1.3 depicts the relationships between these stages.

Feature Extraction consists in computing through audio analysis a feature vector comprising content-descriptors that characterize each song of the music collection [60]. Alternatively, feature vectors can be retrieved from external sources. Songs can be compared using their respective feature vectors, provided a similarity measure has been defined. It is assumed that songs with similar feature vectors are perceptually similar.

The set of feature vectors that corresponds to the entire music collection is a high-dimensional space. Though it is possible to decide if two songs are similar or not based on a similarity measure, visualizing similarity on a high-dimensional space is challenging. **Organization** takes place in order to map the high-dimensional feature space into two or three dimensions using a projection (or dimensionality reduction) technique. The objective of this technique is to arrange the feature vectors in two or three dimensions so that neighbouring vectors on the display are similar and distant vectors dissimilar [54]. Therefore, after organization, perceptually similar songs will be located close to each other on the display, whereas dissimilar songs will be distant from each other.

Self-Organizing Map (SOM) has been widely used to reduce high-dimensional spaces in MIR. Principal Component Analysis (PCA) and Multidimensional Scaling (MDS), however, have been also employed, e.g., *MusicBox* by Lillie [33] (Figure 2.3) was based on PCA and Stober and Nürnberger’s *MusicGalaxy* [54] relied on MDS. SOM is employed in SoundAnchoring because of the optimal use of small screen space on mobile devices. By choosing suitable parameters for the SOM algorithm,

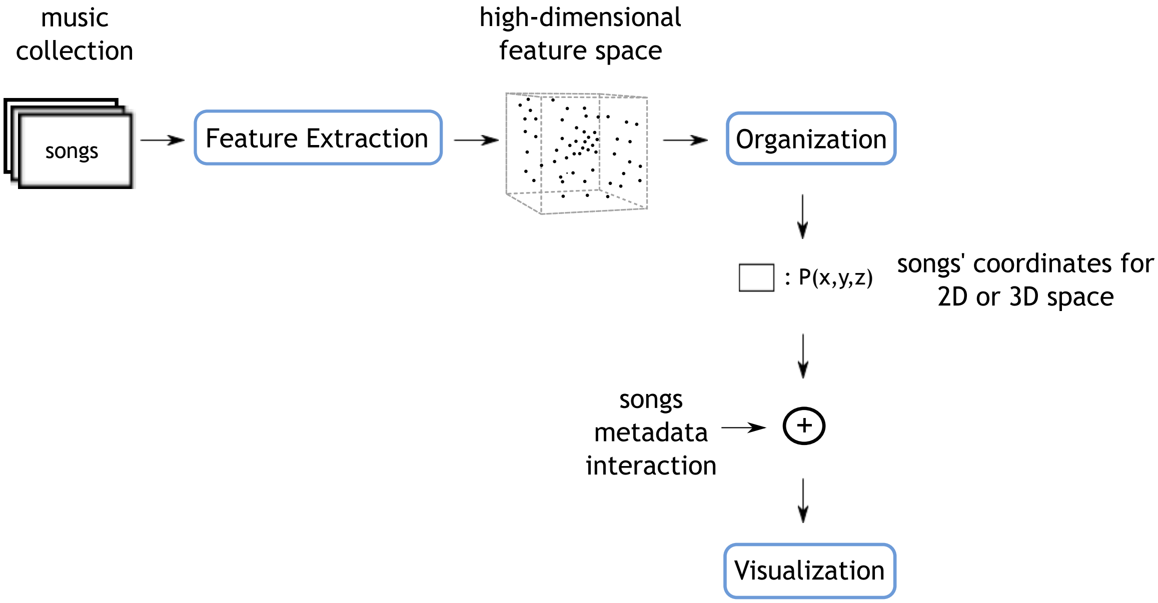


Figure 1.3: Outline of a content-based interface for music collection exploration. During feature extraction, feature vectors that characterize the songs' contents are computed through audio analysis. The feature vectors constitute a high-dimensional space that is mapped to a 2D or 3D space during the organization stage using a dimensionality reduction technique. This technique preserves the topology of the high-dimensional space, i.e., songs whose feature vectors are similar will be close to each other in the 2D or 3D space. Songs that are dissimilar will be located in different areas of the low-dimensional space. The combination of the coordinates of the songs in the low-dimensional space, audio tracks and interaction gestures provided by an API (Application Programming Interface) results in the interactive visualization of the music space. Metadata is usually employed to enrich the interface for music collection exploration.

the SOM grid can display the music space on the screen in an aesthetic way and minimize the occurrences of regions completely devoid of songs. Tolos et al. [57] and Muelder et al. [40] showed that the reduced music space produced by PCA presented problems regarding the distribution of songs. Mörchen et al. [39] suggested that since the output of PCA and MDS are coordinates in a 2-dimensional plane, it is hard to recognize groups of similar songs, unless these groups are clearly separated.

Lastly, **Visualization** consists in displaying the reduced space and providing users with tools to interact with the music collection. The visualization should be customizable. That is, users should be allowed to explore the music library from a variety of vantage points.

With these stages in mind, this thesis focuses on the organization and visualization steps. The following section outlines SOM. It also describes the problem and the research questions that will be handled in the current work.

1.3 Problem statement and research questions

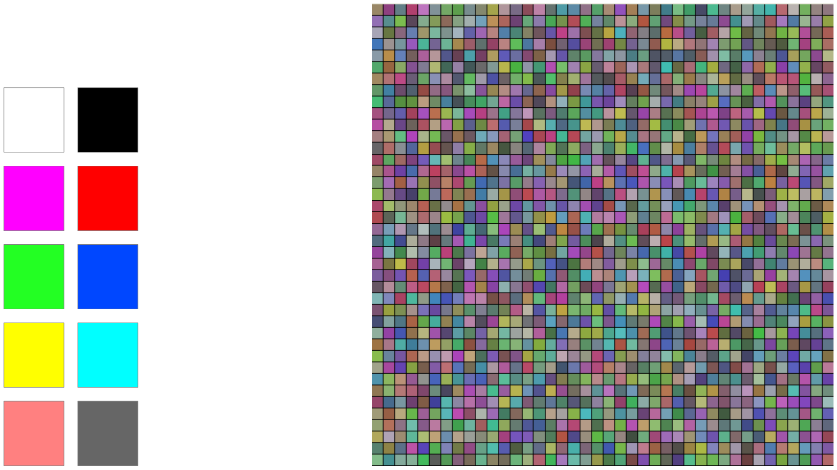
As mentioned in Section 1.2, SOM is the dimensionality reduction technique employed in SoundAnchoring. SOM [28, 29], introduced by Teuvo Kohonen, is an unsupervised neural network that projects high-dimensional data into lower dimensional spaces while trying to preserve the topology of the high-dimensional space. In addition to MIR, SOMs have been used in diverse areas such as automatic speech recognition, cloud classification, micro-array data analysis, document organization, and image retrieval [7].

The traditional SOM is a single layer network that consists of nodes arranged in a 2-dimensional rectangular grid. Nodes (or neurons) have the ability to self-organize based on input vectors (or patterns). During the execution of the SOM algorithm, the neural network is trained with input vectors iteratively, so that different parts of the network become optimized to respond to certain input patterns. The SOM bears similarities with the cerebral cortex in which each region handles different sensory information.

Figure 1.4 depicts a set of inputs and a SOM. Each input vector has three dimensions, each one corresponding to a RGB (red, green and blue) component. Each node of the SOM is associated with a 3-dimensional weight vector. The colour of the node is determined by the dimensions of the corresponding weight vector (RGB components). Weight vectors are initialized with random values, hence the appearance of the SOM.

The SOM has to learn how to represent the 3-dimensional input vectors into the 2D space. The input vectors are presented to the network iteratively and cause different parts of the network to become specialized in responding to certain input vectors, i.e., colours. Figure 1.5 illustrates the building of these clusters of specialized nodes while the traditional SOM algorithm is run.

From a topological point of view, each node is characterized by a position in the 2-dimensional space and a weight vector of the same dimension as the input vectors. Weight vectors are usually initialized with random values. When an input vector is presented to the network, the node whose weight vector is the most similar to the input vector is determined. The input vector is mapped to that node, which is called



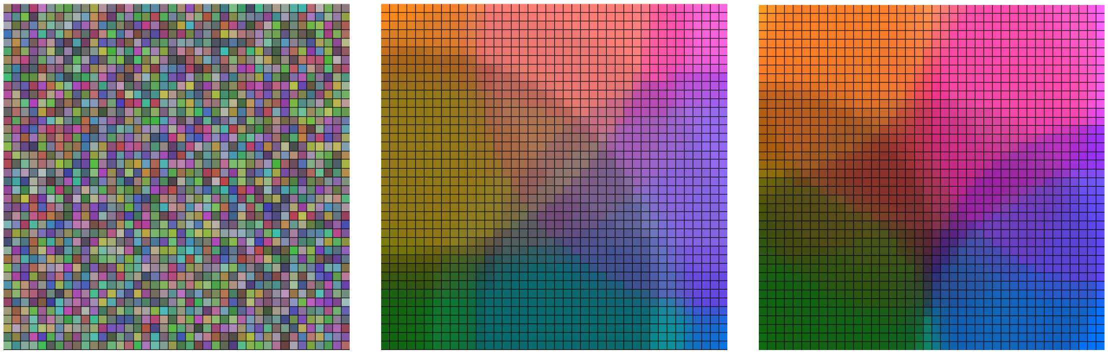
(a) input vectors (b) SOM initialized with random values

Figure 1.4: 3-dimensional input vectors are presented to the SOM. Each node of the SOM is characterized by a position on the grid and a 3-dimensional weight vector which is initialized with random values. The colour of each node is determined by the values of the weight vector, i.e., the RGB components.

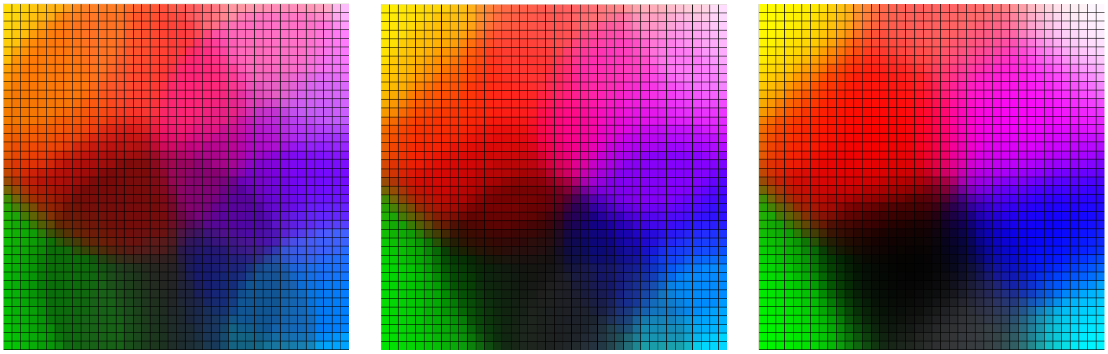
best matching node (BMN). The BMN's weight vector is updated to resemble the input vector. Weight vectors of the BMN's neighbouring nodes are also updated to a certain extent. After several iterations, neighbouring parts of the network will have similar weight vectors. Consequently, these neighbouring parts will respond similarly to certain input patterns.

Since the weight vectors of the SOM are initialized with random values, it is not possible to predict where these specialized regions will be on the grid. Considering SOM-based interfaces for music collection exploration, a user would not be able to determine the positions of clusters containing perceptually similar songs before running the traditional SOM algorithm. Therefore, the user would have to perform an exploratory task to determine the locations of the aforementioned clusters whenever the algorithm is executed.

In Chapter 4, the traditional SOM algorithm is formally explained and a novel technique named **anchoring** is introduced with a view to minimizing the variation in clusters' positions. Anchoring also enables users to personalize the music space by selecting the positions where clusters containing perceptually similar tracks will be located. *SoundAnchoring*, a content-based interface featuring this technique, is designed for this research. Furthermore, SoundAnchoring is evaluated through a user study.



0 iterations 500 iterations 1,000 iterations



1,500 iterations 2,000 iterations 5,000 iterations

Figure 1.5: Snapshots of the SOM during the execution of the traditional algorithm. As the algorithm is executed, different parts of the network will have similar weight vectors. Therefore, these parts will respond similarly to certain input patterns, i.e., colours.

With the user study I aim at answering two research questions:

1. Can anchoring improve the quality of the playlists created?
2. Can anchoring improve the overall perception of the interface for music collection exploration?

1.4 Thesis outline

This thesis is organized as following: Chapter 2 details related work on interfaces for music collection exploration, use of colours in such interfaces and user studies for evaluating interfaces. Chapter 3 presents information on the music collection used in SoundAnchoring. Moreover, the chapter outlines the feature extraction process and the content descriptors employed in the interface. Chapter 4 describes the traditional

SOM algorithm, refines the problem description, and introduces the anchoring technique to address the unpredictability in the locations of clusters containing perceptually similar songs. Chapter 5 provides a thorough description of SoundAnchoring, including design choices regarding colours and interaction. Chapter 6 describes the user study conducted to evaluate SoundAnchoring. Furthermore, the chapter presents information on the participants and the results of the user study. Chapter 7 closes the thesis with conclusion and recommendations for future work.

Chapter 2

Related work

This chapter presents information on research into music collection exploration interfaces that had a bearing on SoundAnchoring's design and evaluation. The first section focuses on the evolution of interfaces for music collection exploration that employed SOMs. The second part places emphasis on the use of colours in visualizations of music spaces. Finally, the third section describes user studies that were conducted to assess music collection exploration interfaces.

2.1 SOM in music collection exploration

Several content-based interfaces have employed SOMs to generate visualizations of music collections. This section aims at telling the abridged story of SOMs in music collection exploration interfaces. It does so by highlighting the metaphors for depicting the music space and the improvements in interaction with music collections.

SOMeJB or SOM-*extended Jukebox*, devised by Rauber and Frühwirth [50], introduced SOMs in music collection exploration but still relied heavily on text to represent the music space. SOMeJB extended the functionalities of the SOMLib digital library system [51], which could organize a collection of text documents according to their content. The visualization of music collections produced by SOMeJB comprised a grid displaying the names of the songs grouped according to acoustic similarities. Even though SOMeJB represented a major departure from metadata-based organization, text was still the prevalent element in the interface, as seen in Figure 2.1. It is worth emphasizing that Cosi et al. [11] and Feiten and Günzel [19] had already employed SOMs to organize sounds from instruments in clean and degraded conditions, and sounds recorded from a sample synthesizer, respectively. SOMeJB, however, was the first system that used SOM to organize a music collection.

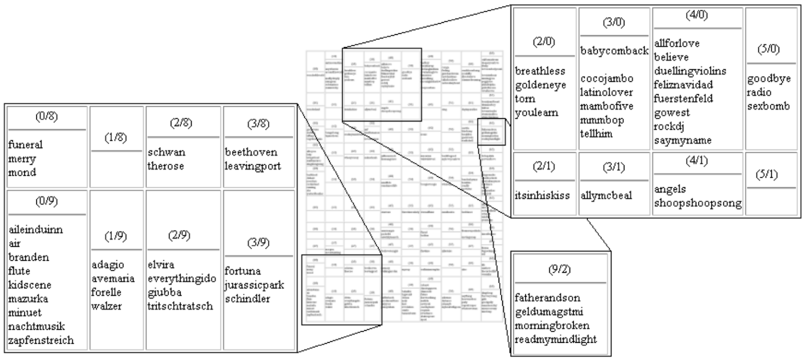


Figure 2.1: SOMeJB [50]: music collection exploration interface derived from the SOMLib digital library system [51]. Songs were grouped according to similarity by the SOM unlike interfaces based on contextual descriptors. SOMeJB's visualization of the music space, however, still relied heavily on text. Image © Andreas Rauber, 2001, by permission.

With *Islands of Music*, developed by Pampalk et al. [46, 48], the importance of text in SOM-based interfaces starts to diminish. Pampalk et al. re-designed the feature extraction process employed in SOMeJB and introduced a new visualization metaphor based on geographic maps named Islands of Music. Clusters containing similar songs corresponded to islands. Songs that could not be mapped to any of the islands were placed on the sea. Connections between clusters were represented by isthmuses. 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. Figure 2.2 shows a music collection visualized using Islands of Music.

Islands of Music inspired several content-based interfaces that also employed SOMs. In addition to employing the island metaphor, these interfaces refined the possibilities of interaction between users and music collections. *PlaySOM*, developed by Neumayer et al. [41], used essentially the same geographic metaphor of Islands of Music but featured additional functionalities with regard to user interaction. In PlaySOM, users could add all the songs of a SOM node to the playlist or select songs by drawing trajectories on the map. The latter mechanism is similar to one of the methods for building playlists implemented in SoundAnchoring.

SOMeJB, Islands of Music, and PlaySOM's approaches to music collection exploration hinged solely on visual communication between the interface and the user.

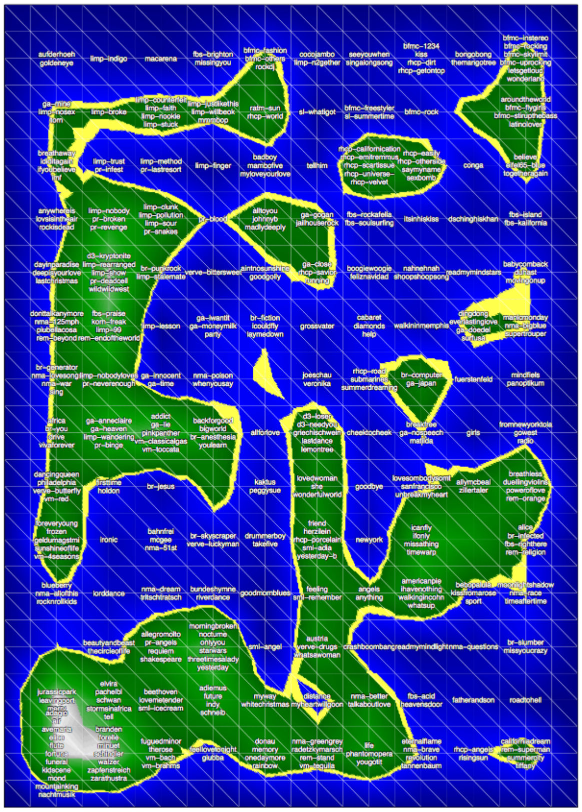


Figure 2.2: In Islands of Music [46, 48], a geographic metaphor was used to represent the music space. Perceptually similar songs were clustered into islands. Mountains and hills corresponded to sub-clusters. Similar clusters were connected by isthmuses and diverse clusters separated by the ocean. Image © Elias Pampalk, 2001, by permission.

Therefore, they did not make use of the human capabilities of processing sound information. The "cocktail party effect", which refers to "the ability to focus one's listening attention on a single talker among a cacophony of conversations and background noise" [2], exemplifies these capabilities.

Brazil et al. [5, 6] had already investigated combinations of visual and auditory communications for sound collection exploration. 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 yet spatially arranged according to their distances to the cursor. User studies conducted later revealed that browsing music collections with audio playback increased the user's efficiency and satisfaction levels [1].

Further refinements in interfaces for music collection exploration using SOMs were based on auditory information. *Sonic SOM*, devised by Lübbers [34], combined a SOM-based visualization with spatial music playback derived from Brazil et al.'s previous research to improve the user's exploration experience. With a view to providing users with an immersing experience, Knees et al. [27] developed *nepTune*, which is essentially a 3D version of Islands of Music. Users would navigate through the music collection with a game pad while songs close to the listener's current position were played using a 5.1 surround system. Semantic information from the Web, e.g., tags and artist-related images, would be displayed on screen to describe the song being played. Lübbers and Jarke [35] conceived an interface similar to nepTune. Auditory feedback was refined by attenuating the volume of the songs that deviated from the user's focus of perception. Clusters containing similar songs would correspond to valleys separated by hills. The aforementioned interfaces featured auditory feedback during the exploration of music collections, which is also implemented to a certain extent in SoundAnchoring.

The perception of music is highly subjective. Consequently, listeners employ different methods to explore their music collections. In addition to several possibilities of interaction and auditory feedback, interfaces should ideally adapt to the user's behaviour. Even though the development of user-adaptive interfaces is still incipient in MIR, some implementations are worth mentioning. The previously described work of Lübbers and Jarke [35] allowed users to customize the environment by changing the positions of the songs, adding landmarks, building or destroying hills, i.e., by modifying the similarity model employed to organize the music collection. The system would then re-build the environment to reflect the user's preferences.

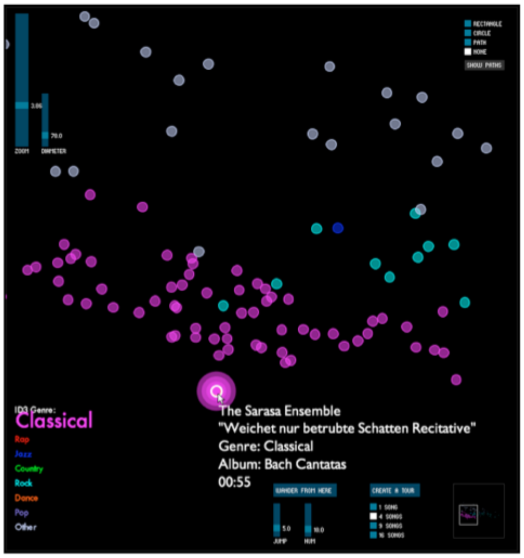
Similar approach was adopted by Stober and Nürnberger [55], who developed *BeatlesExplorer*, shown in Figure 2.4. In this interface, a music collection comprising 282 Beatles songs was organized in hexagonal cells using SOMs. A user could drag and drop songs between cells, which would cause the system to re-locate other songs so that the collection organization would satisfy the user's needs. Likewise, SoundAnchoring features an anchoring mechanism that allows users to customize the location of clusters containing similar songs in SOMs. The anchoring mechanism constitutes a step towards the development of user-adaptive interfaces for music collection exploration.

The increase in processing power and storage for mobile devices and new possibilities in user interaction provided by touch-based interfaces motivated the development of interfaces for music collection exploration as well. *PocketSOMPlayer* [41], created by Neumayer et al. in 2005, was an interface derived from PlaySOM and geared towards mobile devices. Songs could be added to a playlist by drawing trajectories on the SOM. Later the user could remove songs from the playlist created using a text-based screen. Further improvements in interaction brought by multi-touch gestures stimulated the design of interfaces that allowed visually-impaired people to interact with music collections without relying on the WIMP (window, icon, menu, pointer) paradigm. In the prototype developed by Tzanetakis *et al.* [59] for iPhones, for example, a random song would begin to play as soon as the user touched a square on the SOM grid. Moving one finger across squares would cause songs from adjacent squares to cross-fade with each other, thereby generating auditory feedback necessary for assistive browsing of music collections. The same mechanism for producing auditory feedback is implemented in SoundAnchoring. The next section presents information on the use of colours in interfaces for music collection exploration.

2.2 Colours in music collection exploration

Colour has been used in several ways in interfaces for music collection exploration. Even though there seems to be a tendency to map different colours to diverse moods, genres, or styles, other associations such as brightness (or saturation) with song density have been employed as well.

Since colours are excellent for labelling and categorization [64], the SOM-based Islands of Music [46,48] and related interfaces such as PlaySOM [41] and nepTune [27], employed colours to depict clusters and sub-clusters of perceptually similar songs: yellow (beach), dark green (forest), light green (hills), grey (rocks) and white (snow). In order to separate clusters, dark blue (deep sea) and light blue (shallow water) were used. In i3DMO or *Interactive 3D Music Organizer*, developed by Azcarraga & Manalili [3], coloured spheres represented SOM nodes. Spheres would have different colours depending on the genres of the songs mapped to them. Coloured strips on the surface of a sphere would mean the sphere had songs of different genres. MusicBox, designed by Lillie [33], also used hard-coded colour-genre associations, as seen in Figure 2.3. In *Musicream*, conceived by M. Goto and T. Goto [24], songs were represented by discs of different pastel colours, according to the mood conveyed by the piece of music. Similarity in colour would correspond to close moods. *MusicRainbow*, devised by Pampalk and Goto [47], placed artists' names on a circular rainbow. Each colour of the rainbow corresponded to a different style of music.



The Sarasa Ensemble
"Weichet nur betrubte Schatten Recitative"
Genre: Classical
Album: Bach Cantatas
00:55

Figure 2.3: MusicBox [33]. PCA was used to reduce the high-dimensional feature space into 2D coordinates. Colours conveyed information on genres. The colour-genres associations were hard-coded. Image © Anita Lillie, 2008, by permission.

Apart from genre, mood, and style, colours have been also employed to communicate other types of information. In *BeatlesExplorer* [55], seen in Figure 2.4, SOM nodes were emerald green and brightness was used to convey information about the sizes of the nodes. The SOM-based prototype developed by Tzanetakis et al. [59] employed saturation to provide information on the node's song density, i.e., nodes were coloured darker or lighter according to the number of tracks that were mapped to them. It is worth mentioning that some papers on interfaces for music collection visualization present scant information on the use of colours.

In SoundAnchoring, colours are used to convey information on different genres. Since research showed that there is no basis for universality of any genre-colour associations [26], users are allowed to build genre-colour associations with seven palettes containing harmonious colours. The use of colour palettes gives users some freedom to create genre-colour mappings and may have a positive bearing on the aesthetics of visualizations of the music space. The palettes employed in SoundAnchoring were derived from Eisemann's work [17]. Eisemann associated groups of colours to abstract categories such as capricious, classic, earthy, playful, spicy, warm, etc. The aforementioned categories refer to moods that each colour grouping evokes when utilized advertisements, images or documents. The colours of each grouping created by Eisemann were chosen from the Pantone Matching System, a *de facto* colour space

Figure 2.4: BeatlesExplorer [55], an interface for exploration of The Beatles' discography based on SOM. The nodes were coloured in emerald green and brightness was used to convey information on the nodes' sizes. Image © Sebastian Stober, 2010, by permission.

standard in publishing, fabric and plastics [12]. The following section describes user studies that influenced the assessment of SoundAnchoring.

2.3 Interface evaluation

The number of user studies conducted to assess interfaces for music collection exploration is quite limited since the MIR focus has been primarily systems-centric [65]. The importance of user studies to MIR is, however, incontestable, as they allow researchers to evaluate if MIR concepts and systems can be applied to real-world scenarios. This section provides information on evaluations of interfaces for music collection exploration that had a bearing on the design of the user study described in this thesis. The descriptions emphasize the type of tasks participants had to perform, data collected, and include participants' gender and academic background.

Bossard et al. [4] developed a content-based interface that allowed users to visualize 10-dimensional music spaces using two metaphors. The lens metaphor was employed for detailed visualization of a song and its neighbouring audio space. Distant areas were "blurred out", i.e., the interface would present less information on them. The cake metaphor was used for visualizing clusters of similar songs in terms of music genres. During the user study, nine participants had to build playlists using the content-based interface designed and a visualization interface shipped with smartphones. After interacting with each interface, participants rated each song of the playlist using a 11-point system. Furthermore, subjects used a 5-point scale to rate statements about the interfaces and the playlist. The proposed interface outperformed the commercial alternative in most criteria.

Within the SOM realm, Miller et al. [37] designed and evaluated *GeoShuffle*, an interface for iPhones and iPods. GeoShuffle employed SOM to reduce the dimensionality of the feature space. Self-organizing tag clouds enriched the interface with text-based information. Positioning information provided by the device and user's listening habits were employed to recommend songs. One user study was conducted to assess the use of self-organizing tag clouds for music collection exploration. Participants (fourteen computer science graduate students: three female and eleven male) rated statements using a 5-point scale and results showed the interface was perceived as effective and fun. A single participant used GeoShuffle for three weeks so that the location-aware music recommendations could be evaluated. The quality of the recommendations made by the system was measured based on the number of recommended songs that were skipped by the user. The number of songs skipped was smaller when the location information was used to make recommendations.

Vignoli and Pauws [62] developed a system in which a user could customize the similarity model employed to organize the music collection by assigning different weights to content-based and contextual descriptors. User study participants (seven female and fifteen male) built playlists using the system featuring the customizable similarity model and two control systems: one with limited customization possibilities and one with an immutable similarity model. All systems used the same interface: the *Expressive Music Jukebox*. Participants rated statements regarding the systems using a 7-point scale. Interactions with the systems were logged and analyzed. The user study revealed that the fully customizable system was perceived as the most useful, yet most difficult one. Besides, playlists built with the fully customizable system were better rated than playlists built with the system featuring limited customization possibilities.

Hoashi et al. [25] conducted a user study to evaluate the effectiveness of two content-based interfaces for music collection exploration. The first interface displayed the music collection as a list. By selecting one of the songs of the list, perceptually similar songs would be presented to the user. The second interface made use of a 2D space metaphor to generate a visualization of the entire "universe" of songs according to their similarities. Users could select sub-spaces for more thorough exploration. In the user study, participants had to search for songs performed by specific artists in both interfaces. Participants (sixteen computer science students) also rated statements about both interfaces using a 5-point scale. The time required and the number of songs played to complete the task were used as objective measures. Results of the user study revealed that the visualization interface was better accepted than the list-based interface. It took a long time for users to get acquainted with the visualization interface, nonetheless. A new interface was designed to address this shortcoming. The interface utilized a 3D space metaphor for the visualization of the song collection. Genres were used to label sub-spaces. Furthermore, users could decide which content descriptors would be used to visualize the music space. Another user study was carried out to evaluate if the problems of the 2D visualization had been solved by the new interface. The 3D visualization was better perceived than the 2D one by participants. Moreover, it took participants less time to get acquainted with the 3D interface.

In order to assess SoundAnchoring, a user study is conducted. Participants interact with SoundAnchoring and with a control system that does not feature anchoring. This approach is similar to the one adopted by Vignoli and Pauws. As for tasks, the user study also requires participants to build playlists as did Bossard et al., and Vignoli and Pauws.

The user study done in the context of this thesis collects subjective measures using statements as did all the user studies described in the section. A 6-point scale, however, is used so that participants are required to indicate at least a slight preference as there is no centre option. The interaction of the participants with the interfaces is logged to obtain objective measures, as done by Vignoli and Pauws, and Hoashi et al. Unlike the other works described in this section, the user study conducted to evaluate SoundAnchoring tries to ensure a more representative sampling by balancing gender and inviting individuals from different academic backgrounds to take part in the assessment of the interface.

Chapter 3

Feature extraction

The design of a content-based music collection exploration interface can be divided into three main blocks: Feature Extraction, Organization and Visualization. This chapter describes the extraction of features, and presents information on the music collection and content-based descriptors employed in SoundAnchoring. Subsequent chapters focus on organization and visualization.

3.1 Music collection

With a view to conducting a user study as similar as possible as a music collection exploration done in a real-world scenario, datasets used in MIR research were avoided. A collection comprising 700 songs of 10 different music genres (classical, country, dance, electronic, hiphop, jazz, pop, r&b, reggae and rock) was used in SoundAnchoring. All songs were initially in MP3 format with bit rates of 320 kbps. Before the feature extraction, some file processing steps took place. Firstly, audio tracks were converted to .wav format. In order to avoid lead in and lead out effects, 700 30-second audio clips were produced using the files in .wav format. These audio clips were employed during feature extraction.

3.2 Content-based descriptors

Section 1.1 outlined contextual and content-based descriptors. Contextual descriptors, or metadata, present objective or subjective concepts about a piece of music, such as track name, duration, genre, and mood [10]. Content-based descriptors are directly extracted from the audio signal and provide information on “what the music sounds like” [33].

Fu et al. [21] propose a classification system based on music understanding in which content-based descriptors are divided into low-level and mid-level features. Low-level features are directly obtained from signal processing techniques, whereas mid-level features are extracted on top of low-level ones. Neither low nor mid-level features provide information on how listeners interpret and understand music. Top-level labels, such as genre, mood, and style, are closely related to how music is perceived by individuals.

In order to build the content-based interface evaluated in this thesis, the following low-level features were extracted from the music collection: 13 Mel-Frequency Cepstral Coefficients (MFCCs), Spectral Centroid, Spectral Rolloff and Spectral Flux. Descriptions of each feature are presented below.

MFCCs

MFCCs have their origins in speech processing research. The Mel scale was derived from human listening tests, which showed that the frequency intervals producing equal increments in perceived pitch get wider as the frequency increases [53]. Therefore, MFCCs try to simulate the human auditory system.

Spectral Centroid

The spectral centroid C_t is the centre of gravity of the magnitude spectrum of the short-time Fourier transform (STFT):

$$C_t = \frac{\sum_{n=1}^{N} M_t[n] * n}{\sum_{n=1}^{N} M_t[n]} \tag{3.1}$$

in which $M_t[n]$ is the magnitude of the Fourier transform at frame t and frequency bin n. The spectral centroid conveys information about the “brightness” of a sound.

Spectral Rolloff

The spectral rolloff is the frequency R_t below which 85% of the magnitude distribution is concentrated:

$$\sum_{n=1}^{R_t} M_t[n] = 0.85 * \sum_{n=1}^{N} M_t[n] \tag{3.2}$$

The spectral rolloff is a measure of spectral shape.

Spectral Flux

The spectral flux F_t is the squared difference between the normalized magnitudes of successive spectral distributions:

$$F_t = \sum_{n=1}^{N} (N_t[n] - N_{t-1}[n])^2 \tag{3.3}$$

in which $N_t[n]$ and $N_{t-1}[n]$ are the normalized magnitudes of the Fourier transform at the current frame t and the previous one $t-1$, respectively. The spectral flux is a good measure of the amount of local spectral change.

3.3 Extracting the features

Through feature extraction the content of a song is translated into a sequence of numbers known as feature vector. In SoundAnchoring, each feature vector comprises 64 features. The feature vectors of the entire music collection are used as input to the SOM algorithm.

In this thesis, features were extracted using *Marsyas* [58], an open-source audio processing framework. 23-millisecond **analysis windows** were employed to capture the short-time behaviour of the sound. The sound "texture", however, required the computation of running means and standard deviations of the extracted features over a **texture window** of 1 second, which corresponded to forty-three analysis windows. This computation resulted in forty-three 32-dimension feature vectors per audio clip. Lastly, calculating the mean and the standard deviation across the entire audio clip yielded one 64-dimensional feature vector. Before being input to the SOM algorithm, the extracted features were normalized between 0 and 1 across the entire music collection. Feature extraction was carried out on a desktop machine and the resulting feature vectors saved in text files.

MFCCs, spectral centroid, spectral rolloff and spectral flux constitute a timbral texture feature vector that hinges on standard features employed in music speech discrimination and speech recognition. The main motivation for using the aforementioned timbral texture feature vector is its good performance in audio retrieval and classification tasks. The use of texture window results in more accuracy in genre classification tasks as shown by Tzanetakis and Cook [60] in their seminal paper.

Chapter 4

Organization

Organization refers to the use of a dimensionality reduction technique to map the high-dimensional feature space into 2 or 3 dimensions. The technique preserves the topology of the high-dimensional space as much as possible, i.e., songs that have similar feature vectors should be placed close to each other and songs that have dissimilar feature vectors should be apart in the low-dimensional space.

This chapter presents a formal description of the traditional SOM algorithm and the problem this thesis explores, namely the variation in the locations of clusters containing perceptually similar songs. Moreover, a modification to the SOM algorithm, termed anchoring, is introduced to minimize the problem.

4.1 SOM: virtues and flaws

This section builds on the outline presented in Section 1.3 to provide a more thorough explanation of the traditional SOM algorithm and the problem addressed by this thesis. Emphasis is placed on the dimensionality reduction of feature vectors extracted from a music library.

Supposing a S-song music collection, each song is represented by a feature (or input) vector F_j of Q dimensions:

$$F_j = [f_0, f_1, f_2, ..., f_{Q-1}], \quad 0 \leq j < S, \tag{4.1}$$

in which Q is the number of extracted features used to describe the musical content of each song. The features f_0, f_1, ..., f_{Q-1} are normalized, i.e., present values between 0 and 1. The set of feature vectors F_j constitutes the high-dimensional input space.

Each node has a position in the SOM $P = [x, y]$ and a weight vector whose dimension is also Q. Therefore, for a N-node network there will be n weight vectors W_k of Q dimensions:

$$W_k = [w_0, w_1, w_2, ..., w_{Q-1}], \quad 0 \leq k < N \tag{4.2}$$

The weights of each node, w_0, w_1, ..., w_{Q-1}, are randomly chosen between 0 and 1.

The feature vectors are then presented to all nodes from the network. A randomly chosen feature vector F_j is compared to all weight vectors using a distance measure d. If the Euclidean distance is employed, the distance between F_j and W_k can be written as:

$$d = \sqrt{\sum_{c=0}^{Q-1} (f_c - w_c)^2} \tag{4.3}$$

d expresses how close the feature vector and the weight vector are. The node whose weight vector is the closest to the input vector (smallest d) is called **best matching node** (BMN). The feature vector F_j, which corresponds to one song of the music collection, is mapped to the BMN. The weight vectors of the BMN and the neighbouring nodes are then adjusted to resemble the feature vector more closely:

$$W(t+1) = W(t) + l(t)\theta(t)[F(t) - W(t)], \tag{4.4}$$

in which t represents the time-step, $l(t)$ and $\theta(t)$ are the learning and the influence functions, respectively. $l(t)$ decays over time, which allows the algorithm to converge. $\theta(t)$ ensures that the effect of the learning is more pronounced for nodes closer to the BMN and non-existent for distant nodes.

$l(t)$ can be written as:

$$l(t) = L_0 e^{-t/\tau} \tag{4.5}$$

L_0 corresponds to the initial learning rate and τ is a time constant.

$\theta(t)$ can be expressed as:

$$\theta(t) = e^{\frac{-d^2}{2\sigma^2(t)}} \tag{4.6}$$

$\sigma(t)$ is the neighbourhood function given by:

$$\sigma(t) = \sigma_0 e^{-t/\tau} \tag{4.7}$$

σ_0 is the initial size of the BMN neighbourhood. The neighbourhood size shrinks over time, as shown in Figure 4.1, which depicts the BMN in red and the neighbouring nodes in cyan.

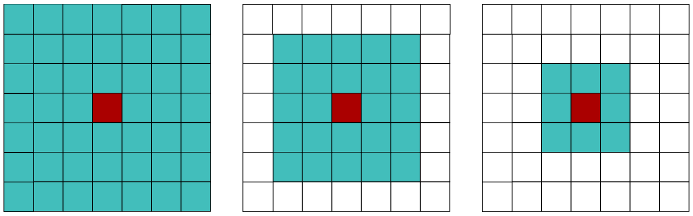


Figure 4.1: BMN (in red) and neighbouring nodes (in cyan). During the training mode, the input vectors are presented to the network. The BMN is the node whose weight vector is the most similar one to the input vector presented to the network. The BMN and the neighbouring nodes are updated to become more similar to the input vector. The adjustments in the weight vectors are more pronounced for nodes close to the BMN. As the algorithm progresses, the size of the neighbourhood shrinks, as shown in the picture.

The algorithm is run iteratively and, in the end, similar feature vectors are placed on the same node or neighbouring nodes, whereas dissimilar feature vectors will be distant from each other. The aforementioned steps refer to the **training mode**, in which the feature vectors of the entire music collection are added to the map. Once the SOM has been trained, new data can be added to the map by determining the best matching node for the new input vector only. This process is called **mapping** or **predicting**.

Figure 4.2 depicts different views of a song collection organized by the SOM algorithm. Songs that sound similar will be close to each other. The SOM algorithm does not have information regarding genre labels as only feature vectors are used as input to the algorithm. The locations of genre clusters are an emergent property of the SOM. The wider the diversity of a genre the more spread out this genre appears on the SOM grid.

Due to the number of random events that take place in the training mode of the traditional SOM algorithm, every SOM generated will be different, even if the same input vectors are considered. With regard to music collection exploration, even if the same feature vectors are used, the positions of the songs in the resulting maps

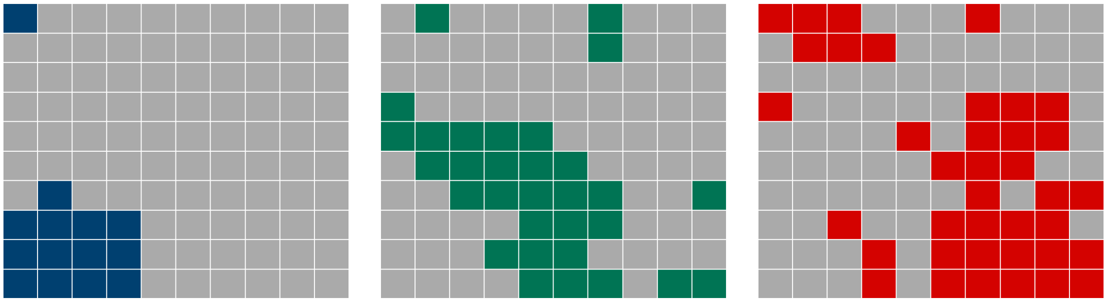

(a) classical songs (b) country songs (c) rock songs

Figure 4.2: Topological mapping of musical content produced by the SOM algorithm: clusters containing (a) classical, (b) country and (c) rock songs. Genre labels are not input to the SOM algorithm. The genre clusters are an emergent property of the SOM.

will be different. The user's underlying understanding of the information or **mental map** [16, 38] is not preserved between executions of the traditional SOM algorithm. Consequently, the user has to re-learn the position of clusters containing similar songs after execution of the traditional SOM algorithm. Figure 4.3 shows variations in the position of a cluster containing classical songs in three maps generated with the same dataset.

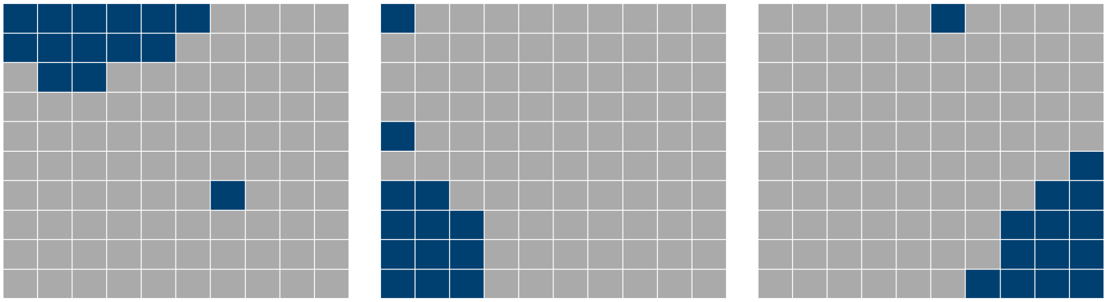

(a) 1st execution (b) 2nd execution (c) 3rd execution

Figure 4.3: Positions of the cluster containing classical music after 3 executions of the traditional SOM algorithm. The same dataset was employed in all executions. Variations in the positions of clusters are a consequence of the initialization of weight vectors with random values. Users of interfaces for music collection exploration based on the traditional SOM algorithm cannot choose the locations of the clusters containing similar songs on the grid.

One may argue that the training mode would be required only once and new songs would be added via mapping, which means the positions of the clusters would remain constant. Considering the current scenario in which millions of tracks are available online, the size of a personal music collection can change dramatically over a short period of time. Therefore, users would have to run the traditional SOM algorithm more often to make the music space visualization reflect the ever changing music libraries more faithfully. Variations in the positions of clusters containing perceptually similar songs would be observed every time the traditional algorithm is run. These variations may have a major bearing on the user experience. Section 4.2 presents a modification to the traditional SOM algorithm to ameliorate the variations in the positions of clusters.

4.2 Anchoring

Due to the nature of the traditional SOM algorithm, it is impossible to know the position of the songs on the map in advance. Clusters with similar songs will have their positions changed every time the traditional SOM algorithm is executed. This section presents a novel technique called **anchoring** that modifies the traditional algorithm SOM to ameliorate the problem described. The modified algorithm is called **anchoredSOM** in this thesis.

Suppose interface users could choose a small number of songs that characterize their music collections. These songs are termed **anchor songs**. Users would also choose the positions of anchor songs on the grid: each anchor song would be located in a different node of the grid. Nodes that receive anchor songs are named **anchor nodes**. Each anchor song would attract similar songs. Consequently, users would be able to determine the positions of clusters containing similar songs in contrast to what happens when the traditional SOM algorithm is run. In order to make this scenario feasible, the anchoredSOM algorithm firstly creates areas around each of the anchor nodes with weight vectors similar to the anchor songs' feature vectors. These areas will attract songs similar to the anchor songs. Later, the anchoredSOM algorithm inputs to the SOM the entire feature vector set and the anchor songs' feature vectors alternately. Songs similar to the anchor songs will form clusters around them. The presentation of anchor songs' feature vectors to the SOM stabilizes the areas around anchor nodes, i.e., keeps the similarity between anchor songs' feature vectors and weight vectors of nodes surrounding the anchor nodes high.

The anchoredSOM algorithm can be divided into four stages, as seen in Figure 4.4:

- **Stage 0.** This stage corresponds to the initialization of the nodes' weight vectors with random values between 0 and 1 in the traditional SOM. In anchoredSOM, nodes' weight vectors are initialized with features vectors randomly chosen from the high dimensional feature space. This approach speeds up the convergence of the SOM algorithm.

- **Stage 1.** This stage consists in presenting only feature vectors of the anchor songs 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 anchor songs are input 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 anchor songs' feature vectors are presented to the SOM. The input of anchor songs' feature vectors for m successive times within one iteration keeps the weight vectors of nodes surrounding anchor nodes similar to the anchor songs' feature vectors.

Figure 4.5 shows the use of anchoring on the classical cluster of the music collection. Anchor songs are represented by small boxes inside the nodes. The first movement of Bach's "Brandenburg Concerto #3" was employed as anchor song and placed on the same node in all executions of the modified algorithm. The variations in the position of the classical cluster were substantially smaller than the ones observed in SOMs without anchoring (cf. Figure 4.3). As shown in Figures 4.6 and 4.7, the technique introduced also gives the users the possibility of placing the clusters in other parts of the grid. Ultimately, anchoring lends itself to personalizing music spaces as the user is now able to choose where clusters containing perceptually similar tracks will be located.

Comparing Figures 4.6 and 4.7, one can realize that the anchoring technique performs better with genres that are distinct and well-localized, such as classical. With genres marked by wide diversity, e.g., pop, the technique still allows users to set the position of the cluster, but songs of those genres will be more spread out on the grid.

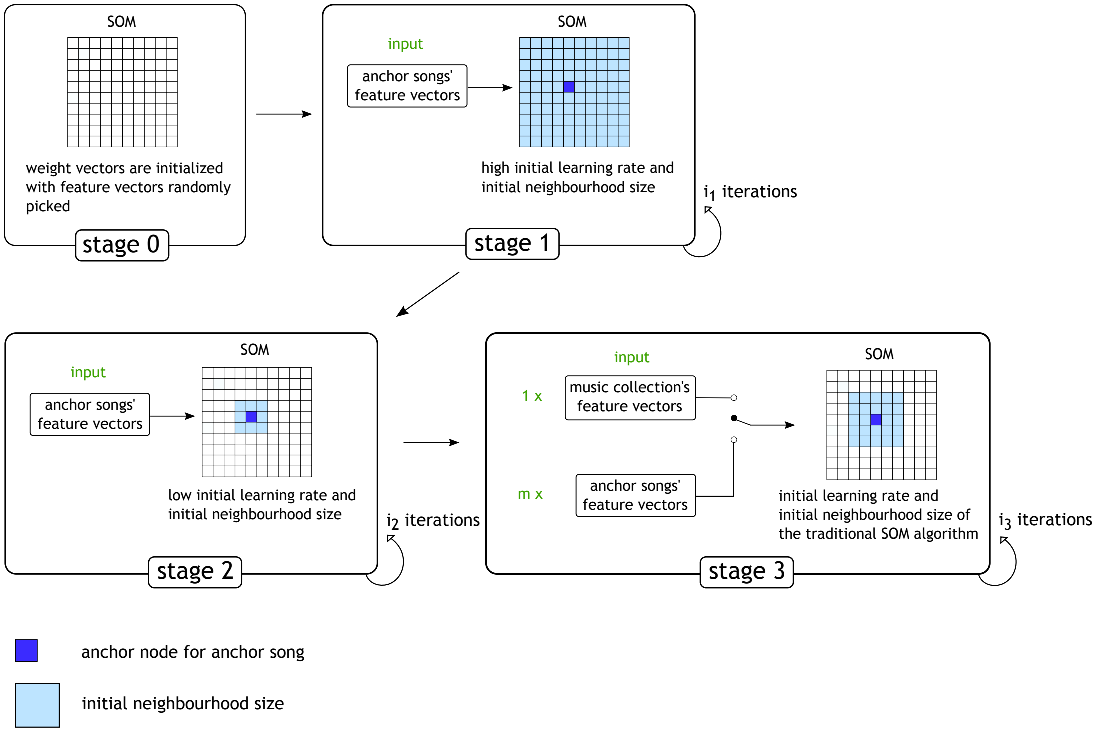
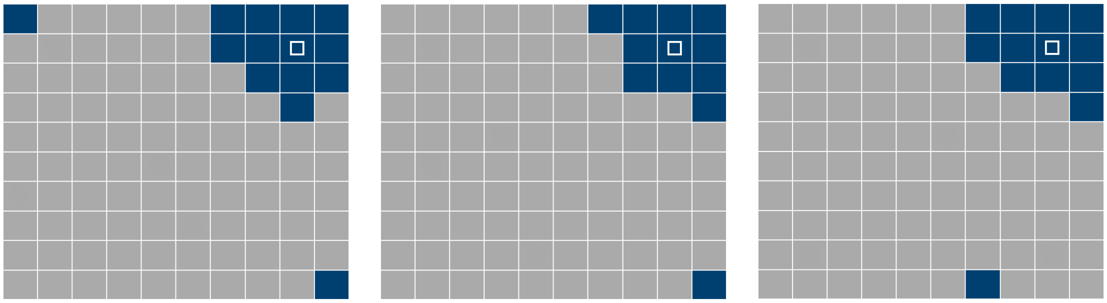
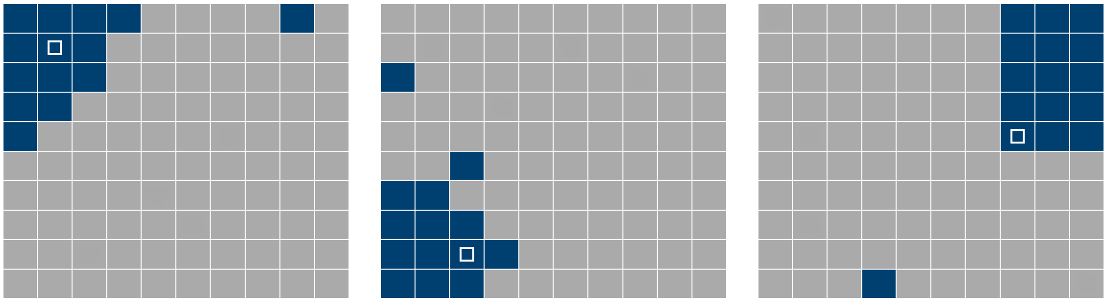


Figure 4.4: The anchoredSOM algorithm comprises four stages. In stage 0, weight vectors are initialized with randomly chosen feature vectors. In stage 1, only anchor songs' input vectors are presented to the SOM with high initial neighbourhood size and learning rate values to cause significant chances to the entire SOM. This process is repeated for i_1 iterations. Stage 2 also uses only anchor songs' input vectors for i_2 iterations but causes small changes in limited areas of the SOM because low neighbourhood size and learning rate values are employed. Stage 3 alternates two sets of input vectors: one comprising the entire music collection and one comprising anchor songs. For each of the i_3 iterations, the input of the entire feature set is followed by m inputs of the anchor songs' feature vectors.



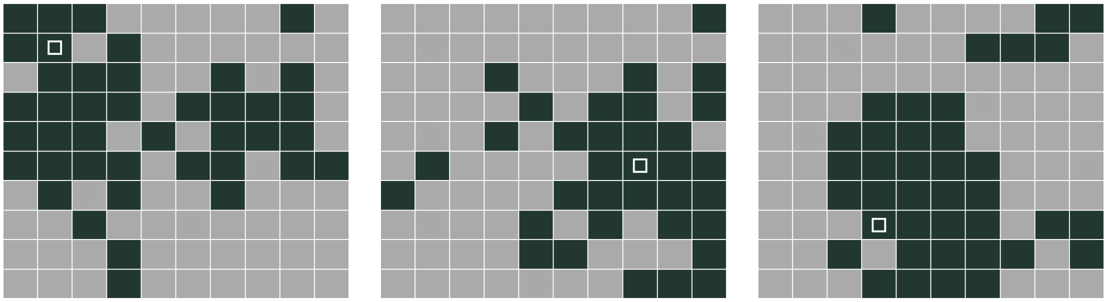
(a) 1st execution (b) 2nd execution (c) 3rd execution

Figure 4.5: Example of anchoring use to preserve the position of a cluster containing classical songs. The first movement of Bach's "Brandenburg Concerto #3", depicted as an inner square with white borders on the top right corner of the grid, was used as anchor song and its position remained constant for three executions of the SOM algorithm. The cluster containing classical music had fewer variations in its position, if compared with SOMs computed without anchoring (cf. Figure 4.3).



(a) 1st execution (b) 2nd execution (c) 3rd execution

Figure 4.6: Example of anchoring use to set different positions for a cluster containing classical songs. The first movement of Bach's "Brandenburg Concerto #3" was used as an anchor song to place the cluster containing classical songs on different areas of the grid.



(a) 1st execution (b) 2nd execution (c) 3rd execution

Figure 4.7: Example of anchoring use to set different positions for a cluster containing pop songs. "Firework" by Katy Perry was used as an anchor song to place the cluster containing pop songs on different areas of the grid.

AnchoredSOM bears similarities with the work of Giorgetti et al. [23], in which SOMs were employed for localization in wireless sensor networks. Giorgetti et al., however, replaced the input vector by the anchor node's weight vector whenever the latter would be chosen as BMN for the former. AnchoredSOM never modifies the input vectors, which correspond to the feature vectors of the music collection. Only weight vectors have their values changed in anchoredSOM.

4.3 Implementation details

Grid size

SoundAnchoring used a SOM grid comprising 100 nodes in a 10x10 configuration. The number of nodes was empirically chosen taking into consideration the number of songs of the music collection dataset. Moreover, it was highly desirable to minimize the number of nodes containing no songs.

SOM parameters

In the user study, participants tested SoundAnchoring and a Control System. SoundAnchoring employed the anchoredSOM algorithm, which is executed after the selection and positioning of anchor songs by the participant. The Control System uses SOMs calculated on a desktop machine with the traditional SOM algorithm, saved in XML (Extensible Markup Language) and loaded during the user study. Parameters for both versions of the SOM were obtained empirically and are presented in Tables 4.1 and 4.2.

parameter	value
initial neighbourhood size (σ_0)	6
initial learning rate (L_0)	0.6
iterations (i)	300

Table 4.1: Empirically-derived parameters for the traditional SOM algorithm used in the thesis.

stage	init. neighbourhood size (σ_0)	init. learning rate (L_0)	iterations (i)	feature vectors input to the SOM
1	10	1	200	anchor songs
2	5	0.5	200	anchors songs
3	6	0.6	500	for each iteration: music collection followed by anchor songs 6 times

Table 4.2: Empirically-derived parameters for the anchoredSOM algorithm.

Number of anchor songs

A pilot study was conducted to determine the number of anchor songs that would be used in SoundAnchoring. Participants were told that an interface able to organize their entire music collections on a 2D grid in a logical manner had been designed. They were also told that information was being collected regarding the number of music genres people needed to organized 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 would use to categorize their collections 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 crayon set.

Later, participants were asked to choose one square of the grid to act as the centre point of each genre. Similar songs 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 anchor songs of different genres.

Chapter 5

Visualization

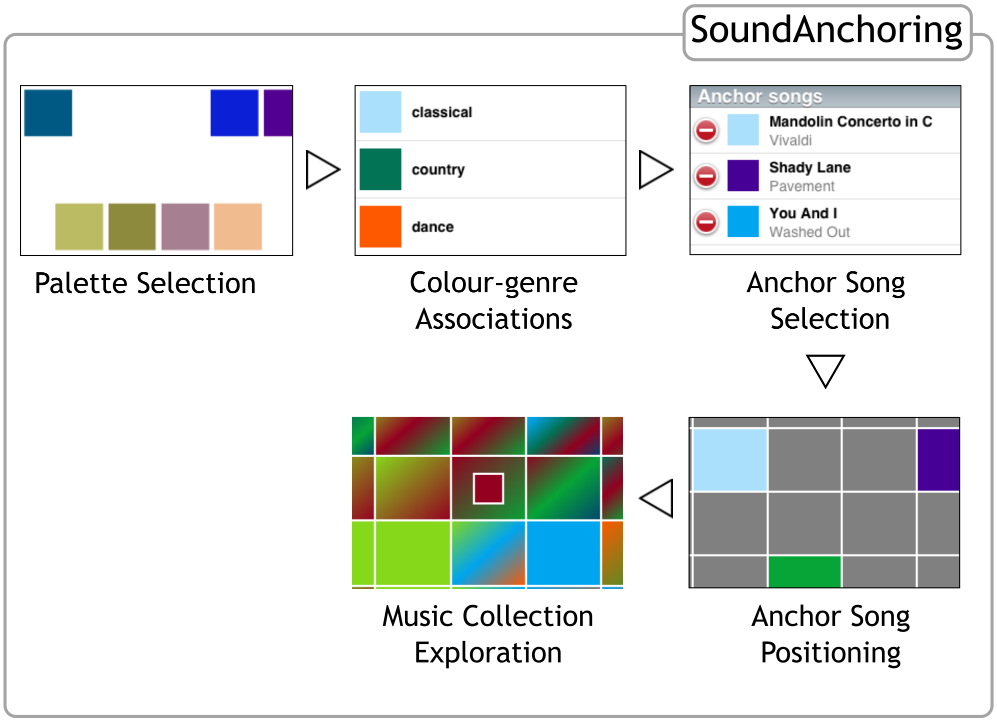


Figure 5.1: SoundAnchoring's sequence of screens.

The visualization stage consists in displaying the output of the anchoredSOM algorithm and providing users with tools to explore the music collection. SoundAnchoring employs the Apple's Cocoa Touch API, which includes gesture recognition and animation, to generate the visualization of the music space. In order to get to the final screen, which contains the music space, users go through a sequence of screens and make choices that will influence the organization and the appearance of the music space. The sequence of screens attempts to lower the cognitive load for the user. Figure 5.1 depicts the aforementioned sequence of screens. Details pertaining to each of the screens are described in the forthcoming sections.

5.1 Palette selection and genre-colour associations

In this work colours convey information on genres. Research suggested that a set of colour-genre mappings that works universally would not not exist [26]. Therefore, the interface designed gives users the opportunity to create their own genre-colour associations.

Even though the interface should be highly customizable and ideally adapt itself to users' patterns of use, some restrictions on colour choices were placed in order to avoid combinations that are aesthetically unappealing. Therefore, genre-colour associations are made via palettes comprising harmonious colours derived from Eisemann [17]. Figure 5.2 depicts the first screen of SoundAnchoring in which participants have to choose one of the seven available palettes to represent the music collection.

Figure 5.2: Palette Selection screen, the first one of the interface: users choose one of the colours palettes to represent the music collection. Palettes were derived from the work of Eisemann [17].

After picking one of the palettes, users build colour-genre mappings by dragging the coloured squares that comprise the chosen palette to the genre names. Faded areas on the palette convey information on colours that have been already mapped to genres. Colours mapped to two different genres can be swapped by dragging one colour to the position occupied by the other colour on the genre list. The screen that allows users to map colours to genres is shown in Figure 5.3.

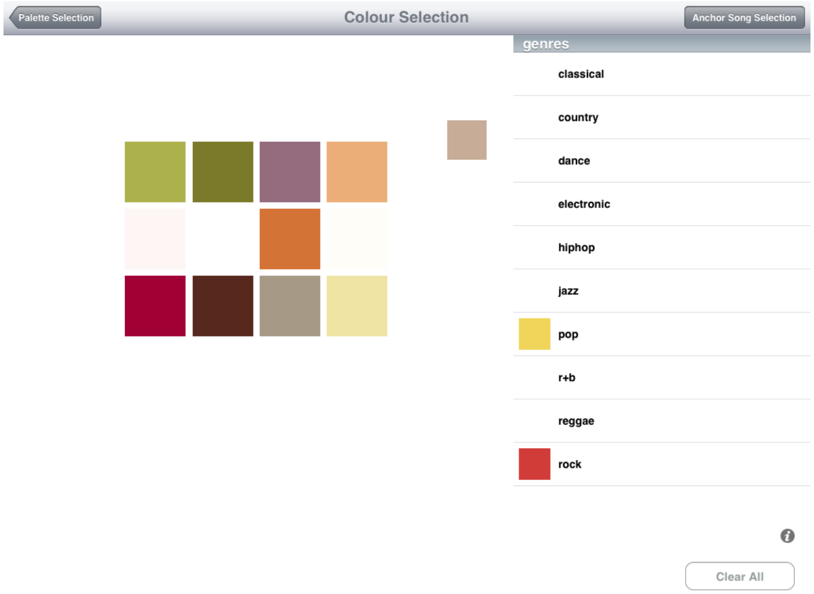


Figure 5.3: Colour Selection, the second screen of SoundAnchoring: users build colour-genre mappings by dragging the colours of the palette to the table containing the genres. Used colours appear faded in the palette. The info button on the bottom right corner provides information on the gestures needed for interaction with the screen elements.

Tapping the information button on the bottom right corner causes a pop over window containing screen-related instructions to appear. Similar buttons are placed on the forthcoming screens to allow access to information without cluttering the interface.

Genres and music collection exploration

Music genres are too vast a topic and can certainly yield several theses. It is, however, relevant to present some information on genres in the context of music collection exploration. These pieces of information motivate the use of genres in SoundAnchoring.

Fabbri [18] defined *genre* as “a kind of music, as it is acknowledged by a community for any reason or purpose or criteria, i.e., a set of musical events whose course is governed by rules (of any kind) accepted by a community".

Classifying music by genre is challenging as genres are influenced by marketing, geographical, historical and cultural factors [44]. There is often considerable overlapping between genres and disagreement arises not only with regard to the genre a song belongs to but also in terms of the label set used for classification [52]. The situation is aggravated by the fact that genre-related information available often refers to artists or albums rather than individual recordings [36].

Given the serious issues presented in the previous paragraph, the question about whether genres should be used in interfaces for music exploration remains. Individuals are used to browsing both physical and online music collections using genres. Furthermore, Apple iTunes and Windows Media, common music player applications, also employ music genre information to organize music libraries. A survey conducted by Lee and Downie [32] revealed that end users are more likely to browse and search by genre than by similar artist or music. A subsequent qualitative study carried out by Laplante [30] found that young adults when searching music for recreational purposes employ genres to filter out undesirable items and, consequently, narrow down the number of items to browse. Therefore, the use of genre information in the interface implemented for the thesis provides users with a familiar vantage point to start exploring music collections.

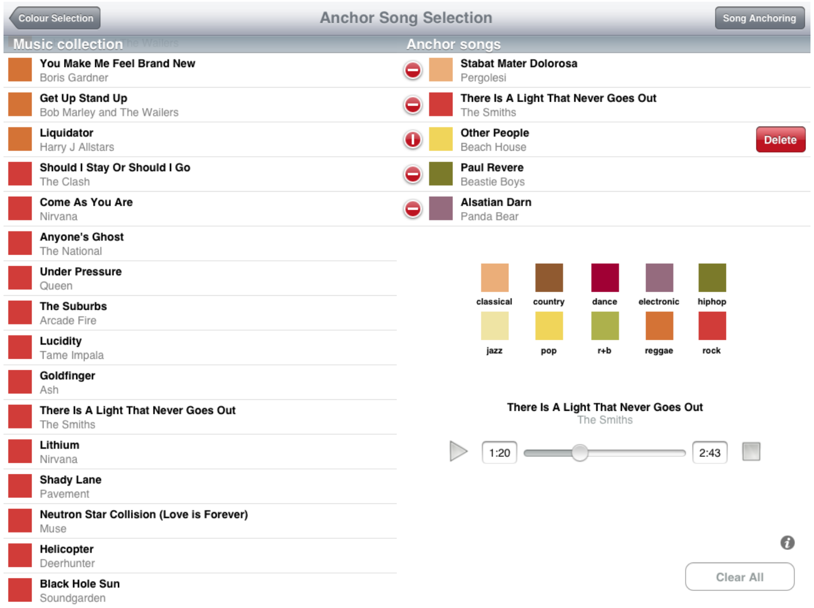
5.2 Anchor song selection and positioning

After choosing the colour palette and making the colour-genre associations, users select anchor songs of different genres. The entire music collection is displayed on a scrollable table. Each row of the table contains the name of the song, the name of the artist and a square coloured according to the genre-colour mapping made on the previous screen. Genres are displayed alphabetically and songs are randomly ordered within a genre.

The genre-colour mappings are presented as buttons. Tapping on one of the buttons, say *jazz*, causes the table to scroll automatically so that jazz songs will be displayed on the screen. This feature facilitates the navigation through long lists of songs.

Tapping once on a row of the music collection table or the anchor song table causes the associated song to start playing and a window displaying a basic music player to appear on screen. The music player displays the names of the song and the artist, and the elapsed and remaining times. The player also features a slider that can be used to listen to specific parts of the song, and control buttons: play, pause and stop. When the stop button is pressed the song stops playing and the window that encompasses the player fades out.

Double-tapping on a row of the table causes the corresponding song to be added to the table that contains the anchor songs. Five anchors songs of different genres have to be chosen. Double-tapping on a song that belongs to the same genre of one of the previously selected anchor songs replaces the latter with the former on the anchor song list. By tapping on any of the“wrong-way” signs, which are part of the Cocoa Touch API, the user initiates the process of removing the song from the anchor song list and causes a delete button to appear. By tapping on the delete button, the user confirms the removal of the song from the anchor song table. Figure 5.4 depicts the anchor song selection screen.



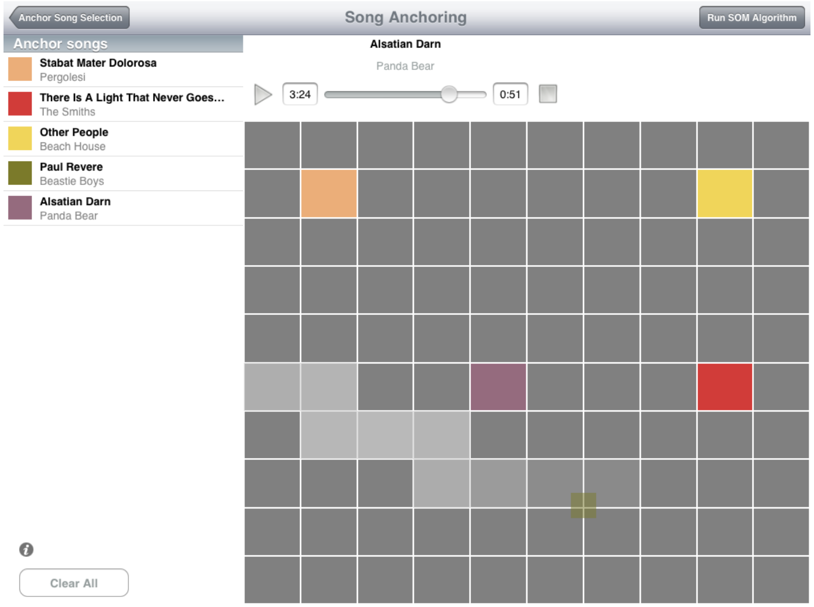
Music collection	
You Make Me Feel Brand New	Boris Gardner
Get Up Stand Up	Bob Marley and The Wailers
Liquidator	Harry J Allstars
Should I Stay Or Should I Go	The Clash
Come As You Are	Nirvana
Anyone's Ghost	The National
Under Pressure	Queen
The Suburbs	Arcade Fire
Lucidity	Tame Impala
Goldfinger	Ash
There Is A Light That Never Goes Out	The Smiths
Lithium	Nirvana
Shady Lane	Pavement
Neutron Star Collision (Love is Forever)	Muse
Helicopter	Deerhunter
Black Hole Sun	Soundgarden

Anchor songs	
Stabat Mater Dolorosa	Pergolesi
There Is A Light That Never Goes Out	The Smiths
Other People	Beach House
Paul Revere	Beastie Boys
Alsatian Darn	Panda Bear

There Is A Light That Never Goes Out
The Smiths

Figure 5.4: Anchor Song Selection, third screen of SoundAnchoring: users choose five anchor songs of different genres from the music collection. The music collection is displayed on a table. Genres are displayed alphabetically and songs are randomly ordered within each genre. Tapping on the any of the genre buttons makes the table scroll to show songs belonging to that genre. Tapping on a row of the music collection table or the anchor song table causes the corresponding song to start playing and a basic music player window to appear. Double-tapping on a row of the music collection table causes the associated song to be added to the anchor song table.

After selecting the anchor songs, users choose their locations by dragging the coloured squares to the grid. Positions of two songs on the grid can be swapped by dragging the square that represents one song to the position occupied by the other song. Figure 5.5 displays the placement of anchor songs on the grid. In the figure the song “Paul Revere” by Beastie Boys is being dragged to the grid. Tapping once on a row of the anchor song table causes the associated song to start playing and the music player window to appear on screen. The anchor songs and their respective positions are employed to generate the SOM via anchoredSOM. The anchor songs will “attract” perceptually similar songs of the music collection.



Anchor songs
Stabat Mater Dolorosa Pergolesi
There Is A Light That Never Goes... The Smiths
Other People Beach House
Paul Revere Beastie Boys
Alsatian Darn Panda Bear

Figure 5.5: Song Anchoring, the fourth screen of the interface: users determine the positions of the anchor songs on the SOM by dragging the coloured squares to the grid. These positions will remain the same throughout the execution of the anchoredSOM algorithm.

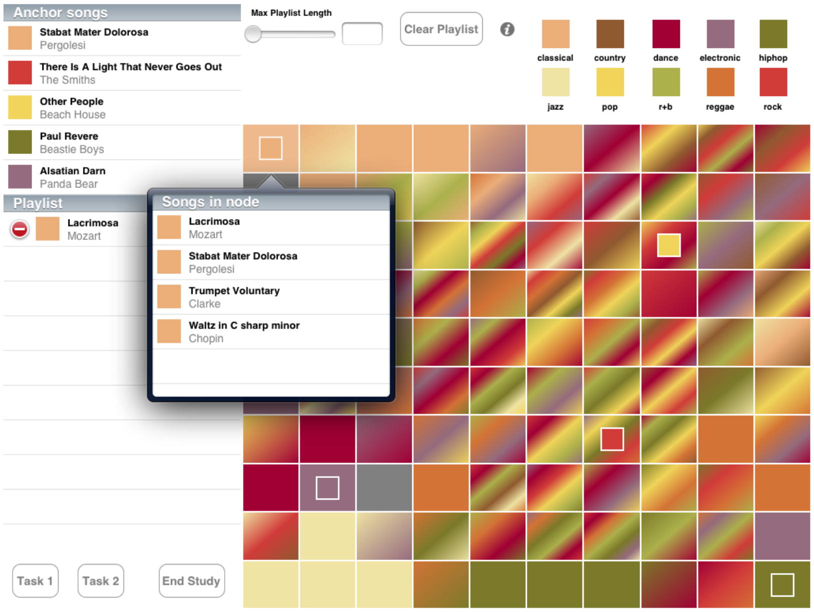
5.3 Music collection exploration

The last screen of SoundAnchoring allows exploration of the music collection. The interface provides users with a visualization of the music space and basic music player capabilities. The main features of the screen are detailed in this section.

Building and editing playlists

SoundAnchoring provides users with gestures to add songs to the playlist, remove them, and change their order as described in the following paragraphs. The playlist is displayed as a table whose rows convey three pieces of the information: genre, by means of a coloured square, song name and artist name.

There are two ways of adding songs to the playlist. By tapping on any node of the grid, users will see a table containing the list of songs mapped to that node by the SOM algorithm. Users can then double-tap on the row that corresponds to a song to add it to the playlist. Figure 5.6 shows the songs mapped to one of the nodes. The track "Lacrimosa" by Mozart has been just added to the playlist.



Anchor songs

- Stabat Mater Dolorosa — Pergolesi
- There Is A Light That Never Goes Out — The Smiths
- Other People — Beach House
- Paul Revere — Beastie Boys
- Alsatian Darn — Panda Bear

Playlist

- Lacrimosa — Mozart

Songs in node

- Lacrimosa — Mozart
- Stabat Mater Dolorosa — Pergolesi
- Trumpet Voluntary — Clarke
- Waltz in C sharp minor — Chopin

Figure 5.6: By tapping on one of the nodes (or squares) of the grid, the user can see the names of the songs mapped to that node organized on a table. Tapping on any of the rows of the table causes the associated song to start playing. Double-tapping on a row adds the corresponding song to the playlist.

Alternatively a user can move one finger across the various nodes of the grid, gesture referred in this thesis as **sketching**. As the user executes this action songs that were mapped to nodes "activated" by the gesture are randomly added to the playlist. The user also receives aural and visual feedback while sketching. Excerpts of the randomly chosen songs cross-fade with each other as the user moves the finger across nodes as a way of providing auditory feedback to users. The opacity of the nodes that have been activated oscillates between two values for a few seconds giving the impression of a trail on the grid. The activated nodes appear faded in Figure 5.7, which depicts the use of the sketching method for adding songs to the playlist.

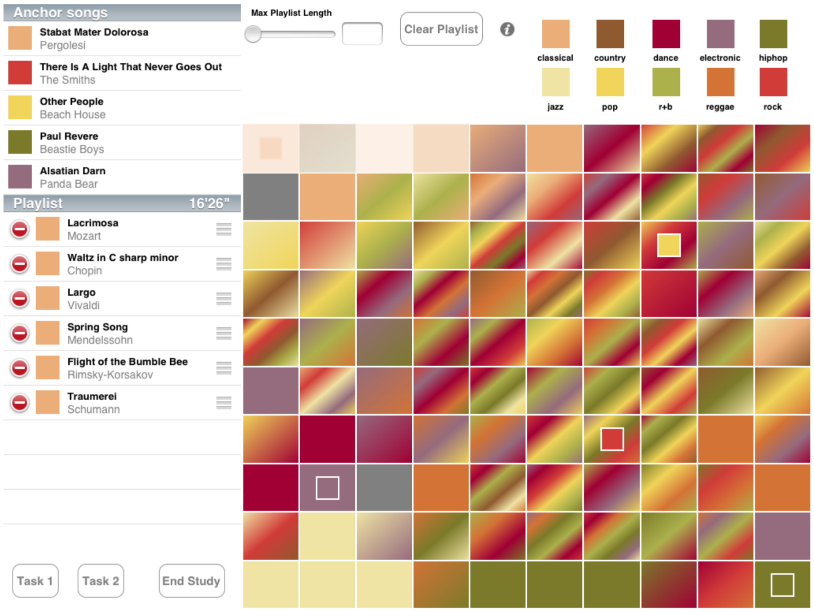


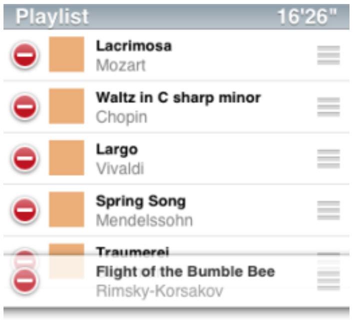
Figure 5.7: Moving one finger across nodes (as if sketching on the surface of the iPad) causes random songs from these nodes to be added to the playlist. Additionally, excerpts from the randomly chosen songs cross-fade with each other providing the user with auditory feedback. The sketching gesture was performed on the faded nodes on the top left corner of the grid.

Adding songs to the playlist individually is similar to the interactions between a user and a text-based interface for music exploration. The SOM, however, ensures that perceptually similar songs will be either in the same node or in neighbouring ones. Adding songs via sketching relies solely on the clustering property of the SOM.

Users can establish the playlist duration by moving a slider located on the top of the screen. As soon as the set duration is reached, a warning window pops on the screen. Users reorder songs via a drag-drop gesture and remove songs using the API metaphor described in Section 5.2. Figure 5.8 illustrates song reordering.

Genres and genre masks

Genres constitute a familiar vantage point to explore music collections as seen in Section 5.1. Genre masks refine the aforementioned vantage point by allowing users to visualize subsets of the music collection. Genre buttons coloured according to the genre-colour associations made by the user are placed on the top right corner of the



Playlist 16'26"

Lacrimosa
Mozart

Waltz in C sharp minor
Chopin

Largo
Vivaldi

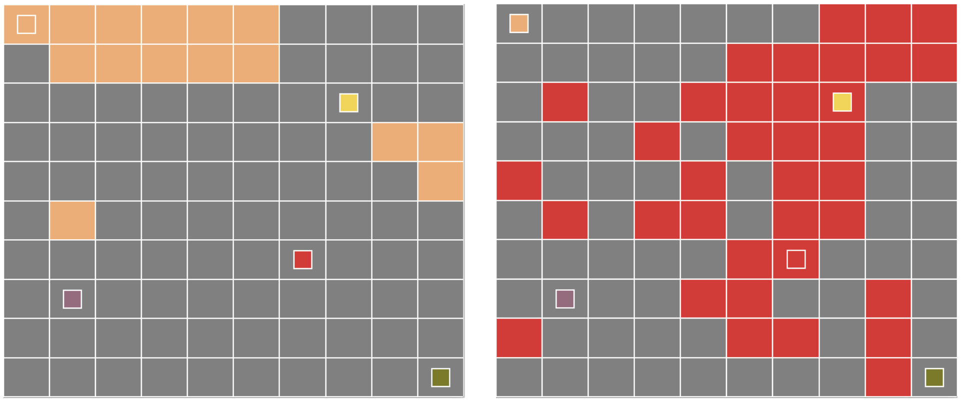
Spring Song
Mendelssohn

Traumerei

Flight of the Bumble Bee
Rimsky-Korsakov

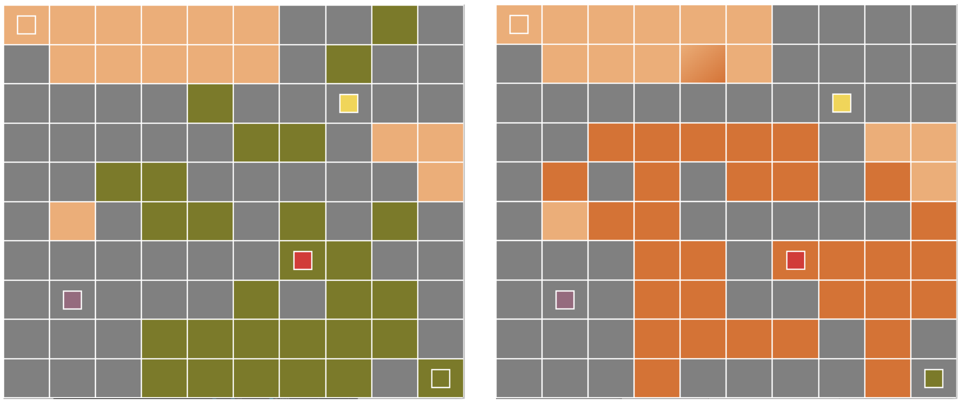
Figure 5.8: Reordering songs using the API metaphor.

music exploration screen. Genres can be "activated" or "deactivated" by tapping on the genre buttons. Deactivating a genre makes both the colour assigned to that genre and the songs belonging to the genre disappear from the map. Consequently, songs from the deactivated genre are not listed when the user taps on a node. Furthermore, sketching across nodes does not add songs from the deactivated genre to the playlist. Therefore, activating or deactivating genres increases or decreases the music space users can interact with. Figures 5.9 and 5.10 depict the use of the genre mask functionality. Figure 5.11 shows details of the interface.



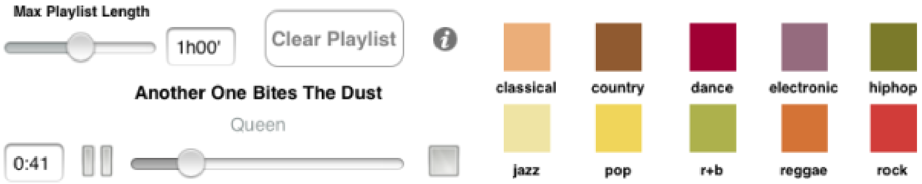
(a) classical songs (b) rock songs

Figure 5.9: Genre masks used individually to depict nodes containing (a) classical and (b) rock songs. Genre masks refine a familiar vantage point to explore music collections, namely genre labels, by allowing users to visualize subsets of the music collection.



(a) classical and hiphop songs (b) classical and reggae songs

Figure 5.10: Genre masks used simultaneously to depict nodes containing (a) classical and hiphop, and (b) classical and reggae songs.



Max Playlist Length 1h00'

Figure 5.11: Interface details. The user can set up the maximum playlist length. The interface will then alert the user by means of a pop over window when the playlist length achieves the threshold set. The basic music player displays the elapsed time/duration of the song being played, as well as the track and artist names. Genre mask buttons allow users to visualize subsets of the music collection filtered by genre labels.

Colours

Colours convey genre information in SoundAnchoring. Solid colours were employed for nodes that contain one or more tracks of the same genre. A gradient technique provided by the Cocoa Touch API is used to colour nodes containing tracks of different genres. The result is aesthetically appealing, as shown in Figure 5.12. The gradients, however, do not inform users of the prevalent genres of each node.

Chapter 6 describes the user study carried out to evaluate SoundAnchoring. The chapter also presents the statistical analysis of measures obtained via user study. Participants' feedback on the interface is also presented.

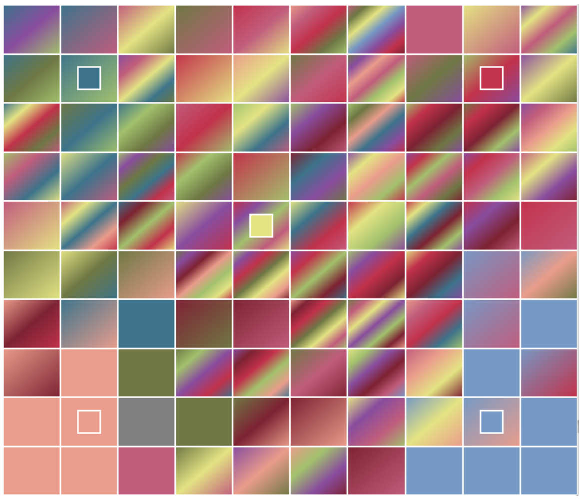


Figure 5.12: Colour scheme adopted to convey information on genres. A colour gradient technique provided by the Cocoa Touch API is employed to colour nodes containing songs from several genres. The locations of anchor songs are shown as inner squares with white borders.

Chapter 6

User study

A user study was conducted to assess SoundAnchoring and verify the hypotheses stated in Section 1.3. The first section of this chapter outlines the user study. The subsequent sections present detailed information on the study and its results.

6.1 Outline of the user study

The user study took place in a prepared office room in December, 2012. Participants were required to use headphones to interact with the interface. Even though a pair of headphones was available, participants were encouraged to bring their favourite pair.

SoundAnchoring and a Control System were loaded in two iPads 2. SoundAnchoring uses the anchoredSOM algorithm and requires participants to choose five anchor songs and their locations on the SOM grid. The Control System loads pre-generated SOMs built with the traditional SOM algorithm and requires users to associate colours and genres only.

Participants were randomly assigned to start working with either SoundAnchoring or the Control System to compensate for order effects. Participants had to perform two tasks on each system. After interacting with each system, subjects rated a set of statements using a 6-point scale. Two versions of the set of statements were used for each subject to minimize acquiescence bias [42]. Subjects wrote about their impressions of each system as well. Participants were also asked to answer a questionnaire regarding their background, music habits, and experience with applications for audio collection exploration and touch-based devices.

Information during the user study

Participants learned that the interface displays songs on a grid according to acoustic similarities between the tracks. This piece of information and a picture of the grid were on the initial screen of both systems. Subsequent screens had also alert windows or "info" buttons to provide participants with extra information unobtrusively. Furthermore, information was also available in printed form and participants were allowed to ask questions at any time.

6.2 Participants

Twenty-one individuals (11 females and 10 males) between 19 and 34 years old (mean: 26.0, standard deviation: 3.67) took part in the user study. Twelve participants have computer science background. The other participants are pursuing undergraduate degrees or have degrees in different areas: philosophy, sociology, geography, biology, veterinary and statistics. Even though none of them is a professional musician, eight participants stated they can play one or more musical instruments. Seventeen individuals stated they listen to music everyday.

With regard to music collection exploration software, all participants stated they only use text-based interfaces. Eleven participants use iTunes, seven use Windows Media Player and four use both applications. Sixteen participants use music applications on touch-screen devices.

6.3 Tasks

After advancing to the last screen of each system, which displays the music space, participants were given the opportunity to explore all the functionalities of the interface before performing the tasks. No time limitations were set for any of the tasks.

The first task was conceived to raise awareness for the mapping of perceptually similar songs to the same node or neighbouring nodes of the SOM. Participants were required to tap on one square of the grid, listen to the songs of that square and songs from adjacent squares. These steps had to be repeated for two other squares, distant from the first square and from each other.

The second task entailed the creation of the 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 30 minutes containing a minimum of 3 genres that would be suitable for the scenario described. The same scenario would be used in both systems (SoundAnchoring and Control System or Control System and SoundAnchoring). Scenarios are described in Appendix A.

6.4 Subjective and objective measures

After trying each system, subjects rated 18 statements using a 6-point scale. Statements were derived from a questionnaire to measure presence in virtual environments [66] and a technology acceptance model questionnaire [13]. Data obtained via statements constituted the subjective measures. Tables 6.1 and 6.2 show mean values for the statements. Higher values are better in Table 6.1, whereas lower figures are better in Table 6.2.

statements	mean rates	
	CS	SA
1. Please rate the playlist you created in task 2.	4.1	**4.2**
2. The interactions with the interface were natural.	3.7	**3.8**
4. The amount of controls available to perform the tasks was adequate.	**4.2**	4.0
5. The auditory aspects of the interface appealed to me.	4.2	**4.3**
8. I felt proficient in interacting with the interface at the end of the experiment.	3.4	**3.6**
10. Getting the system to do what I wanted was easy.	3.8	**4.3**
11. I would consider replacing my current application for music exploration with one based on the system tested.	**3.2**	2.6
15. I enjoyed exploring the music collection with the system.	**4.4**	4.2
16. I can create playlists quickly by using the system.	**3.1**	2.9
18. Please provide an overall rate for the system.	**4.1**	4.0

Table 6.1: Mean values for statement rates for SoundAnchoring (SA) and the Control System (CS). Better rates for each statement are shown in bold.

statements	mean rates	
	CS	SA
3. I was unable to anticipate what would happen next in response to the actions I performed.	1.4	**1.2**
6. The visual aspects of the interface were unappealing to me.	1.0	**0.9**
7. It was impossible to get involved in the experiment to the extent of losing track of time.	1.6	**1.2**
9. The interface was unresponsive to actions I initiated (or performed).	**0.6**	0.8
12. Learning how to use the system was difficult	1.0	**0.8**
13. I disliked creating playlists with the system.	1.0	1.0
14. The system is unsuitable for managing and exploring my music collection.	**1.4**	1.7
17. I disliked the playlists created by using the system.	0.8	0.8

Table 6.2: Mean values for statement rates for SoundAnchoring (SA) and the Control System (CS). Better rates for each statement are shown in bold.

Apart from data collected via statements, interactions of each participant with both systems were translated into 11 objective measures. Table 6.3 presents mean and standard deviation values for objective measures.

6.5 Result analysis

Concerning subjective measures, results show that both systems had similar evaluations with regard to statements concerning control, sensory and distraction factors, which was expected since SoundAnchoring and the Control System have the same interface. In general, participants enjoyed exploring the music collection with the interface and would consider using the interface with their own music collections. Furthermore, subjects liked creating playlists with the interface.

As for objective measures, mean values of objective measures were similar for

objective measures	CS		SA	
	mean	std	mean	std
time spent building playlist (in s)	413.4	211.6	428.9	177.7
songs picked individually	8.4	4.0	8.8	4.3
songs picked by sketching	3.2	7.1	4.5	7.0
songs reordered	3.5	4.2	3.0	4.2
songs picked individually and removed	1.3	2.0	1.3	1.8
songs picked by sketching and later removed	1.0	2.2	1.8	2.5
songs picked from pure nodes	0.6	1.1	0.8	1.8
songs picked from pure nodes and later removed	0.0	0.0	0.2	0.6
songs added to playlist	9.4	4.8	10.2	5.1
nodes tapped	48.9	51.8	50.9	48.9
nodes sketched on	3.9	7.6	4.9	7.6

Table 6.3: Objective measures' mean and standard deviation values for SoundAnchoring (SA) and the Control System (CS).

both systems. One can state that for both systems participants preferred adding songs individually to the playlist to sketching on the interface. The mean value for songs from pure nodes (containing tracks from only one genre) added to the playlist is low for both systems, which suggests pure nodes do not tend to be chosen more often than nodes containing songs of several genres.

On the influence of anchoring

The null hypotheses for this thesis are:

- The anchoring mechanism does not improve the quality of the playlists created.
- The anchoring mechanism does not improve the overall perception of the interface for music collection exploration.

Two statistical techniques were used with the data obtained through the user study: Analysis of Variance (ANOVA) and Fisher's randomization test. The forthcoming paragraphs briefly describe these techniques. Analyses carried out with ANOVA and Fisher's randomization test are also presented.

ANOVA is commonly used in Human Computer Interaction (HCI)-related research [9]. ANOVA is a set of analytic procedures to compare two or more means to check if there are any statistically significant differences among them [56]. Levene's test, which is part of ANOVA, attested no statistically significant differences between SA and CS with regard to overall perception of the interface and quality of playlists ($p > 0.05$). Therefore, both null hypotheses were sustained.

Fisher's randomization test [20] is used to find out if the differences between means or variances observed in data would be observed if the systems under analysis were randomized. Firstly, Fisher's randomization test was employed to compute the statistical significance of the differences between SA and CS using the mean rates for statements 1 to 18. Results of the test are presented in Table 6.4. The difference between the mean rates was statistically significant ($p < 0.05$) for statement 10 ("Getting the system to do what I wanted was easy.") only. This result shows that participants found easier to achieve what they wanted during the music space exploration using SoundAnchoring rather than the Control System. Fisher's randomization test was also used with the mean values of the objective measures. Differences between mean values were not statistically significant as seen in Table 6.5.

Potential learning effects

Three potential learning effects were assessed:

- participants would evaluate the second system better than the first one
- participants with previous experience with touch-screen devices would evaluate the interface better than inexperienced participants
- participants with previous experience with playlist creation would evaluate playlists better than inexperienced participants

Results revealed that learning effects were not statistically significant.

Colour palettes in the user study

Participants of the user study were required to choose one colour palette for each of the systems, SA and CS. Figure 6.1 depicts the colour palettes and the frequency

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

Table 6.4: p-values obtained via Fisher's randomization test for subjective measures. Statistical significance was observed for statement 10 only (in bold).

objective measures	p-values
time spent building playlist (in s)	0.95
songs picked individually	0.42
songs picked by sketching	0.72
songs reordered	0.68
songs picked individually and removed	0.35
songs picked by sketching and later removed	1.0
songs picked from pure nodes	0.32
songs picked from pure nodes and later removed	0.62
songs added to playlist	0.27
nodes tapped	0.75
nodes sketched	0.66

Table 6.5: p-values obtained via Fisher's randomization test for objective measures

each of them was chosen during the user study. Palettes E and F were selected more often, while palette G was never picked by participants of the user study. Table 6.6 shows the choices made by each participant. Nine participants chose the same palette for both systems. Twelve participants selected a different palette for each system.

6.6 Participants' feedback

In addition to rating the 18 statements regarding the playlist and the interface, participants were also asked to write about their impressions of the interface and provide suggestions for its improvement.

The interface

Overall, the interface was favourably reviewed by participants as shown by the words employed to describe it: "intuitive", "easy to use", "aesthetically appealing", "interesting", "flexible", "user-friendly", and "entertaining". More elaborated comments on the interface included: "easy to sample-listen to songs", "it is a fun way to browse

A: 7 times B: 3 times C: 2 times D: 5 times

E: 14 times F: 11 times G: 0 time

Figure 6.1: Colour palettes A-G followed by the number of times each one was chosen during user study.

participant	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
CS	E	F	F	E	D	F	D	B	A	F	D	A	E	F	E	E	F	A	E	B	D
SA	E	E	F	F	B	F	C	A	A	F	A	E	F	E	E	E	A	E	E	C	D

Table 6.6: Palettes chosen by each participant of the user study when interacting with CS and SA.

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”. These comments suggest that participants perceived the visualization of the music collection using SOMs and the clustering of acoustically similar songs as positive.

With regard to colours, participants’ perception was also positive: “I liked the colour blending for visual description of the music collection”, “the colour spectrum of nodes representing which genres were present made it a lot easier to find similar-sounding songs”. Genre masks were considered “useful” and “cool”, but one participant complained that tapping several genre buttons in a short time interval would cause the interface to lag behind.

“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” were some of the statements of the participants regarding playlist creation. As previously mentioned, most participants added songs to the playlist by tapping on each node and selecting the tracks individually, which explains comments like “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, however, particularly liked the sketching gesture for creating 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”.

As for the anchoring mechanism, opinions were, in general, positive. Some participants claimed 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 stated “anchoring didn’t help much”.

Participants stated that the interactions with the interface were “smooth”, “natural" and "perfect for touch screen". Participants that were not familiar with the Apple's metaphor for interacting with files, however, were somewhat confused: "I am used to Windows-based systems and I double-click on files to play them. So it was a little bit hard for me at the beginning to avoid double-tapping on the file when I wanted to play them", "The re-ordering button is unclear as I am not an iPad/iPhone user", "Sometimes I accidentally added a song to the playlist when I just wanted to listen to it".

Suggestions for improvement

Feedback given by participants also included very useful suggestions. Most suggestions refer to minor interface modifications. Others, however, provide avenues for research.

With regard to colours, one participant suggested that colourblind friendly palettes should be added, while another participant would like the interface to suggest a random set of genre-colour associations that could be edited instead of putting the user in charge of building all the mappings from scratch.

A zooming function to explore more thoroughly individual nodes or subsets of nodes of the music space and a search function to locate specific songs on the grid were requested by participants. Another participant would like to have the possibility of adding all the songs of a node to a list with only one gesture. These suggestions show that the interface should incorporate more customization options to cater for different ways of exploring music collections.

The interface ought to be more adaptable to users' behaviour as well. With regard to anchoring, for instance, participants would like the interface to recommend anchor songs based on user's listening habits. Feedback provided suggests that subjectivity is a key word with regard to music collection exploration. Consequently, interfaces should offer more possibilities of customization and learn from users' behaviour.

Chapter 7

Conclusions

The MP3 format, the high availability of audio tracks on the Web, and the decreasing storage prices have facilitated the building of personal music collections comprising thousands of songs. This scenario poses challenges regarding the organization and exploration of music collections.

Popular applications, such as iTunes and Microsoft Media Player, organize the music collection in long lists of text. In these applications, the interaction between the user and the music collection takes place using contextual descriptors, e.g., song name, artist name, genre. While text-based applications are efficient at locating specific tracks, they do not allow listeners to explore music collections with a certain degree of serendipity. Moreover, with text-based applications, users are not able to extract the gist of a music collection without deep exploration.

MIR researchers have designed content-based interfaces for music collection exploration to address the shortcomings of text-based applications. Content-based interfaces organize music collections according to song similarity and generate visualizations of the music space that place perceptually similar songs close to each other. SOM is one of the techniques employed to organize the music collections. Although SOM produces clusters containing similar songs, the SOM algorithm does not allow users to determine the position of these clusters on the music space.

This thesis introduces a variation on the original SOM algorithm named anchoredSOM to allow users to personalize the music space by choosing the positions of clusters containing perceptually similar songs. Consequently, the user's underlying understanding of the music collection organization is preserved between executions of the algorithm. The thesis also presents SoundAnchoring, a content-based interface for music collection exploration featuring anchoredSOM. A user study conducted to assess SoundAnchoring indicates that both the interface and the anchoring technique were favourably rated by participants. The evaluation, however, does not provide evidence for or against the influence of the anchoring technique on the quality of the playlists created or the overall perception of the interface.

Even though the influence of anchoring on the perception of the interface and the quality of playlists could not be quantified, the research conducted makes significant contributions to the MIR field. The first contribution is the interface for music collection exploration featuring the anchoring technique. The second contribution is the evaluation of the interface by means of a user study. Additionally, this thesis brings computational aesthetics closer to MIR by employing colour palettes designed with a view to producing visually appealing results. Lastly, the thesis provides new research routes in MIR and computer graphics detailed in Section 7.1.

7.1 Future work

The process of designing the interface featuring the anchoring technique and the user study carried out to evaluate that interface yielded ideas that could be used to further improve the aforementioned interface. I also believe that this thesis introduces new investigation scenarios for MIR and computer graphics researchers.

Topics for future work were organized using the three blocks introduced to explain the design of content-based interfaces for music collection exploration, namely feature extraction, organization and visualization.

7.1.1 Feature extraction

Extracting content-based descriptors of a music collection is computationally demanding, hence it had to be performed on a desktop computer in this thesis. Research on multicore graphics processing units (GPUs) may lead to methods based on parallelization that would allow feature extraction to be carried out in portable devices, e.g., iPads. Alternatively, content-based descriptors could be obtained from external sources, such as Echo Nest[1].

---

[1] http://echonest.com/

7.1.2 Organization

Dimensionality reduction

SOM was employed to map the high-dimensional set of feature vectors that represent the music collection into a two-dimensional space in this thesis. The original SOM algorithm is computationally expensive. This scenario is further aggravated by modifications to the algorithm that generated anchoredSOM. Research on GPU and parallelization may reduce the computation time of the algorithm. Dividing the organization stage into two steps, e.g., PCA followed by anchoredSOM or even PCA preceded by anchoredSOM, might also yield positive results.

Algorithm evaluation

AnchoredSOM was introduced in this thesis to minimize the variation in the positions of clusters containing similar songs on the SOM grid. Visual and auditory inspection of SOMs produced with the novel algorithm suggests that anchoredSOM allows users to choose the location of the aforementioned clusters before the algorithm is executed. Moreover, the user study conducted showed that anchoring was well received among participants. Further research on anchoring, however, ought to include objective evaluations of anchoredSOM.

7.1.3 Visualization

Research findings obtained through the user study can be used to further improve the user experience:

- **Text-based search.** Users would be able to formulate direct queries using text if they knew exactly what they are looking for.
- **Semantic zoom.** A zoom function, possibly operated by a pinch gesture, would allow users to explore nodes or groups of nodes more thoroughly. The zoom level would determine the amount of information that would be displayed on the screen.
- **Refinements to the playlist-related gestures.** Users would be able to activate or deactivate the addition of songs to the playlist while serendipitously exploring the music collection with the sketching gesture. Furthermore, users

would have the ability to add all songs from a node to the playlist with one single gesture.

- **Suggestions based on listener's history.** Listeners would receive suggestions regarding colour-genre associations and potential anchor songs, based on previous sessions.
- **Colours.** Colour palettes that were not frequently chosen by participants of the user study could be removed or modified. Furthermore, users would also be able to choose colours from different palettes to build customized colour groupings.

A step towards more adaptable interfaces would consist in allowing users to change the locations of anchor songs while exploring the music space. The SOM would then reorganize the song collection on the fly to reflect the changes. If mobile devices are considered for implementation, research into parallelization would be required to ascertain that the interface could remain responsive with the additional workload imposed by the on-the-fly organization.

A modified user study could help assess the impact of anchoring on the perception of the interface and quality of playlists. In this user study, participants would use the systems to perform tasks with their own music collections. Even though both systems would require participants to select anchor songs and their locations on the grid, only one of them would actually execute the anchoredSOM algorithm. Furthermore, the number of participants would be higher. The user study would be more complex and time-consuming as it would entail the normalization of audio tracks and feature extraction of multiple music libraries. Alternatively, participants could have access to the music collection employed in the user study in advance to get acquainted with the tracks.

The development and evaluation of colourblind friendly palettes for genre-colour associations, and the use of the interface to organize pictures using appropriate content descriptors are potential investigation routes related to the computer graphics realm.

Appendix A

Additional Information

A.1 Hardware and software implementation notes

Songs were decoded into .wav using mpg123[1] and trimmed using SoX[2] to generate the 30-second audio clips employed in feature extraction. Features were extracted using Marsyas [58] installed on an iMac 2.7 GHz Intel Core i5 8 GB RAM running OS X 10.7.4. The SOM algorithms and the visualization of the music space were written in Objective-C. Two iPads 2 running iOS 5.1.1 were employed during the user study.

A.2 Performance notes

Feature extraction encompassed the audio analysis of a 700-track library to compute one 64-dimension feature per track and the normalization of the extracted features across the entire music collection between 0 and 1. This stage was performed once using Marsyas [58] running on a desktop computer whose configuration is described in Section A.1. Approximately five minutes were required to complete feature extraction.

The organization stage consisted in calculating the self-organizing map using the anchoredSOM algorithm. This step was carried out on an iPad 2. Approximately two minutes were required to run the anchoredSOM algorithm after the anchor songs and their respective positions on the grid were chosen by the user.

The visualization of the music space entails the combination of the node-track mappings resulted from the anchoredSOM algorithm with the music library and the

---

[1] http://www.mpg123.org/

[2] http://sox.sourceforge.net/

interaction elements of the Apple Cocoa Touch API. The visualization allows users to interact with the music space and customize it according to their preferences. Within the context of visualization, performance is related to the ability of the interface to be responsive to users' actions. Therefore, interaction possibilities that would cause the interface to lag behind and put strain on users were avoided.

A.3 User study

A.3.1 Statements

The statements rated by participants using a 6-point scale were:

1. Please rate the playlist you created in task 2.
2. The interactions with the interface were natural.
3. I was unable to anticipate what would happen next in response to the actions I performed.
4. The amount of controls available to perform the tasks was adequate.
5. The auditory aspects of the interface appealed to me.
6. The visual aspects of the interface were unappealing to me.
7. It was impossible to get involved in the experiment to the extent of losing track of time.
8. I felt proficient in interacting with the interface at the end of the experiment.
9. The interface was unresponsive to actions I initiated (or performed).
10. Getting the system to do what I wanted was easy.
11. I would consider replacing my current application for music exploration with one based on the system tested.
12. Learning how to use the system was difficult.
13. I disliked creating playlists with the system.

14. The system is unsuitable for managing and exploring my music collection.
15. I enjoyed exploring the music collection with the system.
16. I can create playlists quickly by using the system.
17. I disliked the playlists created by using the system.
18. Please provide an overall rating for the system.

A.3.2 Scenarios

Participants built playlists suitable to one of the following scenarios.

- Working in the office
- Jogging
- Romantic dinner
- Working out at the gym
- Celebrating the end of the term
- Driving home after work
- Vaccuming & cleaning the house
- Riding the bus to school to take a final exam
- Car trip to your favourite destination
- Relaxing at home

A.3.3 Data logged

The following items were logged by both systems:

- colour-genre associations
- anchor songs (SoundAnchoring only)
- songs played

- playlist created
- songs added individually to the playlist
- songs added to the playlist by sketching
- songs added individually to the playlist and later removed
- songs added to the playlist by sketching and later removed
- songs of the playlist which were reordered
- songs from pure nodes added to the playlist
- songs from pure nodes removed from the playlist
- time spent building playlist
- number of nodes tapped
- number of nodes sketched on

Appendix B

Lessons learned along the way

Chapters 3, 4 and 5 describe the final choices regarding the design and implementation of SoundAnchoring. These chapters, however, do not present information on the dead ends encountered. The aforementioned predicaments are presented in the forthcoming paragraphs and will hopefully help future MIR researchers.

At the outset it was desired that feature extraction, organization and visualization would take place on the mobile device, i.e., the iPad, as users frequently store their music collections in smartphones, portable music players or tablets. The limited processing power of these devices, however, forced changes in the initial set of requirements.

Feature extraction computes for each track a numerical representation that conveys information on what the track sounds like. Within the context of this research, feature extraction needed to be performed only once and, therefore, was carried out on a desktop machine featuring more processing power and memory than the iPad. Features vectors were saved on a plain text file that could be input to the SOM algorithm.

Initially, the feature vectors (thirteen MFCCs, spectral centroid, spectral rolloff and spectral flux) were calculated directly from a 23-millisecond analysis window. Each track of the music collection would yield a 16-dimension feature vector. This choice was motivated by the influence of the feature vector size on the amount of time required to calculate the self-organizing map. SOMs produced with 16-dimension feature vectors, however, performed poorly with regard to the grouping of acoustically similar tracks. In order to improve clustering, a 1-second texture window was used to capture the long-term behaviour of the sound "texture". The use of a texture window improves the performance of classifiers [60] and, therefore, had a positive bearing on the output of the SOM. The feature set input to the SOM algorithm, however, had now sixty-four dimensions instead of the initial sixteen features.

Running the traditional SOM and the anchoredSOM algorithms using 64-dimension feature vectors as input on a desktop computer was not challenging. Running the algorithms on an iPad, however, would impose long waiting times on users. In order to address this issue, maps used in the Control System were pre-calculated on a desktop computer as they did not require users' input. The anchoredSOM algorithm was lightly modified with a view to reducing the computation time. Firstly, the weight vectors were initialized with feature vectors randomly chosen from the high dimensional feature space. Besides, the squared Euclidean distance was employed when determining the BMN to avoid computation costs regarding the square root operation.

As for visualization, colours were used in different ways before the adoption of the colour scheme featured in SoundAnchoring. The subsequent paragraphs describe these early approaches. Advantages and disadvantages of each attempt are accompanied by figures.

The first colour scheme conceived for visualizing the music collection made use of the weight vectors associated with nodes. Nodes containing anchor songs would be assigned arbitrary colours and the RGB components of the remaining nodes, i.e., the colour of these nodes, would be determined using the weight vectors. Even though this approach could yield aesthetically appealing visualizations of the music collection as depicted by Figure B.1, it only emphasized the properties of the SOM: songs mapped to the same node or neighbouring nodes are similar and songs from distant nodes are perceptually different. No new information on the music collection was conveyed by this colour scheme.

The other two schemes, depicted in Figures B.2 and B.3, used colours to carry genre information. Squares that contained one or more songs of the same genre would have the colour picked by the user for that genre. One question, however, remained unanswered: how nodes that contain songs of two or more genres should be coloured.

The first attempt to address this issue consisted in recursively dividing the square into quadrilaterals. The original square and the nine different mosaics would correspond to nodes containing songs from one to ten different genres. An example of this approach is shown in Figure B.2. The result was not only noisy but also unappealing.

The second attempt also used mosaics to depict squares containing songs of diverse genres. An inner square, however, would convey information on the prevalent genres

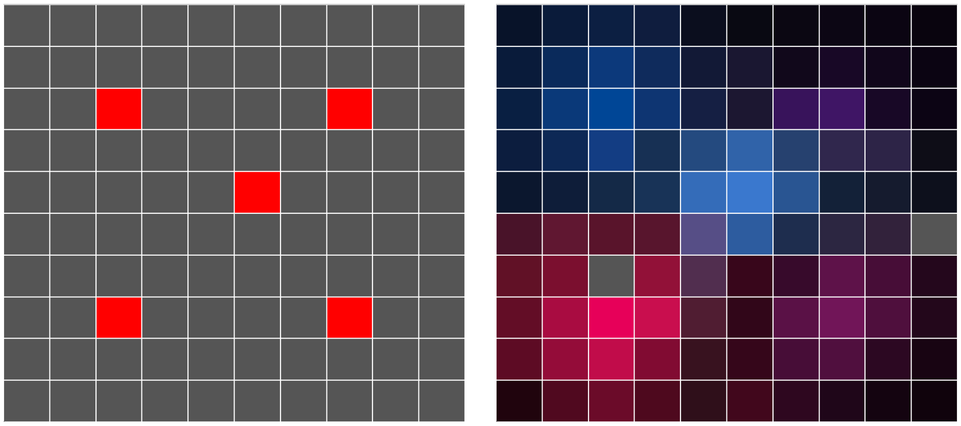

(a) anchor song locations (b) colour scheme

Figure B.1: Colour scheme based on weight vectors. Nodes containing anchor nodes depicted in red (a) were assigned arbitrary colours and the other nodes had their colours determined using the weight vectors (b).

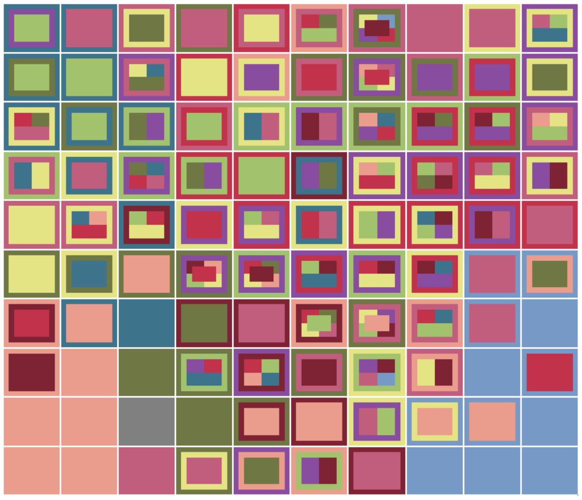


Figure B.2: First attempt at using colours to communicate information on genres. Pure nodes, i.e., nodes containing songs belonging to only one genre had only one colour. Nodes containing songs of different genres presented different colours according to the genre-colour associations built by the user. The result was noisy and the user's cognitive load high.

of the node, while the remaining area of the node would provide information on secondary genres. Figure B.3 depicts a visualization using this method. Although the second visualization was less cluttered than the first one, the user's cognitive load was high due to the communication of information on genres on both axes and the numerous possible combinations of colours and mosaic types.

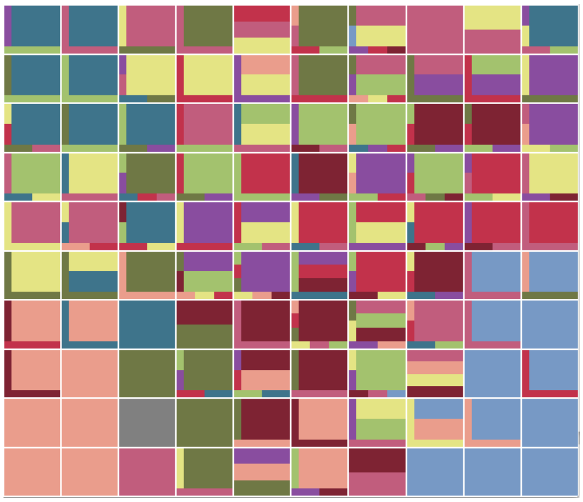


Figure B.3: Second attempt at using colours to communicating information on genres. Pure nodes had one colour. For nodes with songs belonging to two or more genres, an inner square was coloured according to the prevalent genre(s) of the songs mapped to that node. The remaining area of the square communicated information on the other genres of the songs mapped to that node. The scheme was less noisy than the previous one (Figure B.2), yet the user's cognitive load remained high.

With regard to the auditory feedback provided by the interface while users sketched on the grid, the first attempt to implement this feature was made on a first generation iPad. Results were subpar: the hardware was not able to change the opacity of the node and play an audio file simultaneously. Consequently, the first-generation iPad had to be replaced by the iPad 2.

Bibliography

[1] Sarah Ali and Parham Aarabi. A cyclic interface for the presentation of multiple music files. *IEEE Transactions on Multimedia*, 10(5):780–793, 2008.

[2] Barry Arons. A review of the cocktail party effect. *Journal of the American Voice I/O Society*, 12(7):35–50, 1992.

[3] Arnulfo Azcarraga and Sean Manalili. Design of a structured 3D SOM as a music archive. *Advances in Self-Organizing Maps*, pages 188–197, 2011.

[4] Lukas Bossard, Michael Kuhn, and Roger Wattenhofer. Visually and acoustically exploring the high-dimensional space of music. In *International Conference on Computational Science and Engineering*, volume 4, pages 58–65. IEEE, 2009.

[5] Eoin Brazil and Mikael Fernström. Audio information browsing with the sonic browser. In *Coordinated and Multiple Views in Exploratory Visualization, 2003. Proceedings. International Conference on*, pages 26–31. IEEE, 2003.

[6] Eoin Brazil, Mikael Fernström, George Tzanetakis, and Perry Cook. Enhancing sonic browsing using audio information retrieval. In *International Conference on Auditory Display ICAD-02, Kyoto, Japan*, 2002.

[7] Juan C. Burguillo. Playing with complexity: From cellular evolutionary algorithms with coalitions to self-organizing maps. *Computers & Mathematics with Applications*, 2013.

[8] Donald Byrd and Tim Crawford. Problems of music information retrieval in the real world. *Information Processing & Management*, 38(2):249–272, 2002.

[9] Paul Cairns. HCI... not as it should be: inferential statistics in HCI research. In *Proceedings of the 21st British HCI Group Annual Conference on People and Computers*, pages 195–201. British Computer Society, 2007.

[10] Michael A. Casey, Remco Veltkamp, Masataka Goto, Martin Leman, Christophe Rhodes, and Malcolm Slaney. Content-based music information retrieval: current directions and future challenges. *Proceedings of the IEEE*, 96(4):668–696, 2008.

[11] Piero Cosi, Giovanni De Poli, and Giampaolo Lauzzana. Auditory modelling and self-organizing neural networks for timbre classification. *Journal of New Music Research*, 23(1):71–98, 1994.

[12] Gabriela Csurka, Sandra Skaff, Luca Marchesotti, and Craig Saunders. Building look & feel concept models from color combinations. *The Visual Computer*, 27(12):1039–1053, 2011.

[13] Fred D. Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, pages 319–340, 1989.

[14] J. Stephen Downie. Music information retrieval. *Annual review of information science and technology*, 37(1):295–340, 2003.

[15] J. Stephen Downie. Toward the scientific evaluation of music information retrieval systems. In *Proceedings of the 4th International Conference on Music Information Retrieval*, pages 25–32, 2003.

[16] Peter Eades, Wei Lai, Kazuo Misue, and Kozo Sugyama. Preserving the mental map of a diagram. In *Proc. Compugraphics*, pages 24–33, 1991.

[17] Leatrice Eisemann. *Pantone's Guide to Communicating with Color*. Grafix Press, Ltd., Florida, 2000.

[18] Franco Fabbri. Browsing music spaces: Categories and the musical mind. In *Proceedings of the IASPM Conference*, 1999.

[19] Bernhard Feiten and Stefan Günzel. Automatic indexing of a sound database using self-organizing neural nets. *Computer Music Journal*, 18(3):53–65, 1994.

[20] R. A. Fisher. *The Design of Experiments*. Hafner Publishing Company, New York, 1935.

[21] Zhouyu Fu, Guojun Lu, Kai Ming Ting, and Dengsheng Zhang. A survey of audio-based music classification and annotation. *Multimedia, IEEE Transactions on*, 13(2):303–319, 2011.

[22] Joe Futrelle and J. Stephen Downie. Interdisciplinary research issues in music information retrieval: ISMIR 2000–2002. *Journal of New Music Research*, 32(2):121–131, 2003.

[23] Gianni Giorgetti, Sandeep Gupta, and Gianfranco Manes. Wireless localization using self-organizing maps. In *Proceedings of the 6th international conference on Information processing in sensor networks*, pages 293–302. ACM, 2007.

[24] Masataka Goto and Takayuki Goto. Musicream: New music playback interface for streaming, sticking, sorting, and recalling musical pieces. In *Proceedings of the 6th International Conference on Music Information Retrieval*, pages 404–411, 2005.

[25] Keiichiro Hoashi, Shuhei Hamawaki, Hiromi Ishizaki, Yasuhiro Takishima, and Jiro Katto. Usability evaluation of visualization interfaces for content-based music retrieval systems. In *Proceedings of the 10th International Society for Music Information Retrieval Conference*, pages 207–212, 2009.

[26] Jukka Holm, Antti Aaltonen, and Harri Siirtola. Associating colours with musical genres. *Journal of New Music Research*, 38(1):87–100, 2009.

[27] Peter Knees, Markus Schedl, Tim Pohle, and Gerhard Widmer. An innovative three-dimensional user interface for exploring music collections enriched with meta-information from the web. In *Proceedings of the ACM Multimedia*, pages 17–24, 2006.

[28] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69, 1982.

[29] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[30] Audrey 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*, pages 601–606, 2010.

[31] George Lawton. Improved flash memory grows in popularity. *Computer*, 39(1):16–18, 2006.

[32] Jin Ha Lee and J. Stephen Downie. Survey of music information needs, uses, and seeking behaviours: Preliminary findings. In *Proceedings of the International Conference on Music Information Retrieval*, pages 441–446, 2004.

[33] Anita S. Lillie. MusicBox: Navigating the space of your music. Master's thesis, Massachusetts Institute of Technology, 2008.

[34] Dominik Lübbers. SoniXplorer: Combining visualization and auralization for content-based exploration of music collections. In *Proc. of ISMIR*, pages 590–593, 2005.

[35] Dominik Lübbers and Matthias Jarke. Adaptive multimodal exploration of music collections. In *Proc. of ISMIR*, volume 2009, 2009.

[36] Cory McKay and Ichiro Fujinaga. Musical genre classification: Is it worth pursuing and how can it be improved? In *Proceedings of ISMIR 2006, Victoria, Canada*, pages 101–107, 2006.

[37] Scott Miller, Paul Reimer, Steven Ness, and George Tzanetakis. Geoshuffle: Location-aware, content-based music browsing using self-organizing tag clouds. In *International Symposium on Music Information Retrieval*, 2010.

[38] Kazuo Misue, Peter Eades, Wei Lai, and Kozo Sugiyama. Layout adjustment and the mental map. *Journal of visual languages and computing*, 6(2):183–210, 1995.

[39] Fabian Mörchen, Alfred Ultsch, Mario Nöcker, and Christian Stamm. Visual mining in music collections. *From Data and Information Analysis to Knowledge Engineering*, pages 724–731, 2006.

[40] Chris Muelder, Thomas Provan, and Kwan-Liu Ma. Content based graph visualization of audio data for music library navigation. In *IEEE International Symposium on Multimedia (ISM)*, pages 129–136. IEEE, 2010.

[41] Robert Neumayer, Michael Dittenbach, and Andreas Rauber. PlaySOM and pocketSOMplayer, alternative interfaces to large music collections. In *Proc. of ISMIR*, volume 5, 2005.

[42] J. C. Nunnally. *Psychometric Theory.* McGraw-Hill, New York, 1967.

[43] Celma Òscar and Paul Lamere. Music recommendation tutorial. In *Proceedings of the 8th International Conference on Music Information Retrieval*, 2007.

[44] François Pachet and Daniel Cazaly. A taxonomy of musical genres. In *Proc. Content-Based Multimedia Information Access (RIAO)*, pages 1238–1245, 2000.

[45] Ted Painter and Andreas Spanias. Perceptual coding of digital audio. *Proceedings of the IEEE*, 88(4):451–515, 2000.

[46] Elias Pampalk. Islands of music: Analysis, organization, and visualization of music archives. Master's thesis, Vienna University of Technology, Vienna, Austria, 2001.

[47] Elias Pampalk and Masataka Goto. Musicrainbow: A new user interface to discover artists using audio-based similarity and web-based labeling. In *Proceedings of the 7th International Conference on Music Information Retrieval, Victoria, BC, Canada*, 2006.

[48] Elias Pampalk, Andreas Rauber, and Dieter Merkl. Content-based organization and visualization of music archives. In *Proceedings of the 10th ACM International Conference on Multimedia*, pages 570–579. ACM, 2002.

[49] Martin Peitz and Patrick Waelbroeck. An economist's guide to digital music. *CESifo Economic Studies*, 51(2-3):359–428, 2005.

[50] Andreas Rauber and Markus Frühwirth. Automatically analyzing and organizing music archives. *Research and Advanced Technology for Digital Libraries*, pages 402–414, 2001.

[51] Andreas Rauber and Dieter Merkl. The SOMlib digital library system. *Research and Advanced Technology for Digital Libraries*, pages 852–852, 1999.

[52] Mohamed Sordo, Òscar Celma, Martín Blech, and Enric Guaus. 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*, page 255260, 2008.

[53] Stanley Smith Stevens, John Volkmann, and E.B. Newman. A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, 8(3):185–190, 1937.

[54] Sebastian Stober and Andreas Nürnberger. Musicgalaxy - an adaptive user-interface for exploratory music retrieval. In *Proc of 7th Sound and Music Computing Conference*, 2010.

[55] Sebastian Stober and Andreas Nürnberger. Towards user-adaptive structuring and organization of music collections. *Adaptive Multimedia Retrieval. Identifying, Summarizing, and Recommending Image and Music*, pages 53–65, 2010.

[56] Barbara G. Tabachnick and Linda S. Fidell. *Using multivariate statistics.* Allyn and Bacon, Boston, 5th edition, 2007.

[57] Marta Tolos, Raquel Tato, and Thomas Kemp. Mood-based navigation through large collections of musical data. In *Consumer Communications and Networking Conference, 2005. CCNC. 2005 Second IEEE*, pages 71–75. IEEE, 2005.

[58] George Tzanetakis. Music analysis, retrieval and synthesis of audio signals MARSYAS. In *Proceedings of the 17th ACM International Conference on Multimedia*, pages 931–932. ACM, 2009.

[59] George Tzanetakis, Manjinder Singh Benning, Steven R. Ness, Darren Minifie, and Nigel Livingston. Assistive music browsing using self-organizing maps. In *Proceedings of the 2nd International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '09, pages 3:1–3:7, New York, NY, USA, 2009. ACM.

[60] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

[61] Remco Veltkamp. Multimedia retrieval algorithmics. *SOFSEM 2007: Theory and Practice of Computer Science*, pages 138–154, 2007.

[62] Fabio Vignoli and Steffen Pauws. A music retrieval system based on user-driven similarity and its evaluation. In *Proc. International Symposium on Music Information Retrieval*, pages 272–279, 2005.

[63] Edward Walker, Walter Brisken, and Jonathan Romney. To lease or not to lease from storage clouds. *Computer*, 43(4):44–50, 2010.

[64] Colin Ware. *Information visualization: perception for design.* Morgan Kaufmann, 2004.

[65] David M. Weigl and Catherine Guastavino. User studies in the music information retrieval literature. In *Proceedings of the 12th ISMIR Conference*, pages 335–340, 2011.

[66] Bob G. Witmer and Michael J. Singer. Measuring presence in virtual environments: A presence questionnaire. *Presence*, 7(3):225–240, 1998.

[67] Alejandro Zentner. Measuring the effect of file sharing on music purchases. *Journal of Law and Economics*, 49(1):63–90, 2006.