An Innovative Three-Dimensional User Interface for Exploring Music Collections Enriched with Meta-Information from the Web

Peter Knees[1], Markus Schedl[1], Tim Pohle[1], and Gerhard Widmer[1,2]
[1]Department of Computational Perception, Johannes Kepler University Linz, Austria
[2]Austrian Research Institute for Artificial Intelligence (OFAI)
peter.knees@jku.at, markus.schedl@jku.at, tim.pohle@jku.at, gerhard.widmer@jku.at

ABSTRACT

We present a novel, innovative user interface to music repositories. Given an arbitrary collection of digital music files, our system creates a virtual landscape which allows the user to freely navigate in this collection. This is accomplished by automatically extracting features from the audio signal and training a Self-Organizing Map (SOM) on them to form clusters of similar sounding pieces of music. Subsequently, a Smoothed Data Histogram (SDH) is calculated on the SOM and interpreted as a three-dimensional height profile. This height profile is visualized as a three-dimensional island landscape containing the pieces of music. While moving through the terrain, the closest sounds with respect to the listener's current position can be heard. This is realized by anisotropic auralization using a 5.1 surround sound model. Additionally, we incorporate knowledge extracted automatically from the web to enrich the landscape with semantic information. More precisely, we display words and related images that describe the heard music on the landscape to support the exploration.

Categories and Subject Descriptors: H.5.1 Information Interfaces and Presentation: Multimedia Information Systems

General Terms: Algorithms

Keywords: Music Similarity, User Interface, Clustering, Visualization, Web Mining, Music Information Retrieval

1. INTRODUCTION

The ubiquity of digital music is definitely a characteristic of our time. Everyday life is shaped by people wearing earphones and listening to their personal music collection in virtually any situation. Indeed, it can be claimed that recent technical advancements and the associated enormous success of portable mp3 players, especially Apple's iPod, have formed the Zeitgeist immensely. Even if these developments have changed the way we access music, organization of music has basically remained unmodified. However, from the constantly growing field of Music Information Retrieval many interesting techniques to advance the accessibility of music (not only on portable devices) have emerged over the last few years.

With our application, we provide new views on the contents of digital music collections, beyond the uninspiring but regrettably frequently used structuring scheme *artist – album – track*. Our interface offers an original opportunity to playfully explore and interact with music by creating an immersive virtual reality that is founded in the sounds of one's digital audio collection. Using intelligent audio analysis, the pieces of music are clustered according to sound similarity. Based on this clustering, we create a three-dimensional island landscape that contains the pieces. Hence, in the resulting landscape, similar sounding pieces are grouped together. The more similar pieces the user owns, the higher is the terrain in the corresponding region. The user can move through the virtual landscape and experience his/her collec-

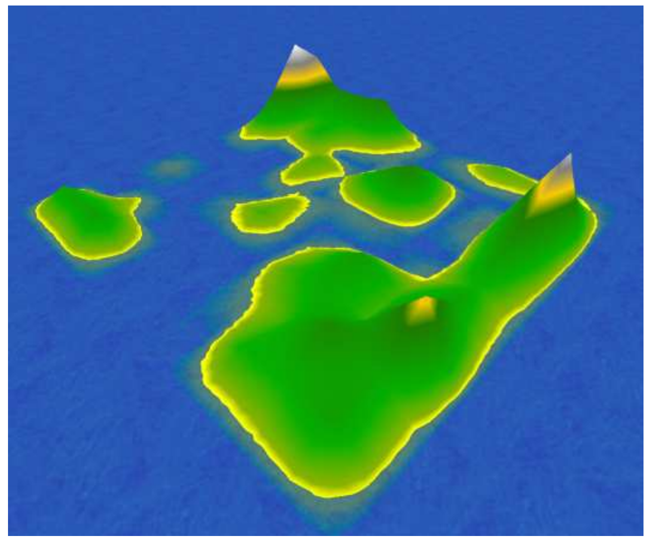

Figure 1: An island landscape created from a music collection. Exploration of the collection is enabled by freely navigating through the landscape and hearing the music typical for the region around the listener's current position.

tion. This visual approach essentially follows the *Islands of Music* metaphor from [16]. Each music collection creates a characteristic and unique landscape. Additionally to seeing the music pieces in the landscape, the pieces closest to the listener's current position are played. Thus, the user gets an auditory impression of the musical style in the surrounding region. To accomplish the spatialized audio playback, we rely on a 5.1 surround sound system.

Furthermore, the system incorporates web-retrieval techniques to enrich the landscape with semantic and visual information. Instead of displaying song title and performing artist on the landscape, the user can also choose to display words that describe the heard music or images that are related to this content. Thus, besides the intelligent and content-based organization of music, the system also accounts for the cultural aspects of music by including additional information extracted from the web.

The remainder of this paper is organized as follows. In the next section, we will give a brief overview of existing alternative interfaces to music archives and preceding work. In Sections 3 and 4, we describe the technical fundamentals and the realization of the application. In Section 5, we report on a small user study we conducted. Finally, we review our interface and propose future enhancements that will further increase the project's practical applicability.

2. RELATED WORK

It is one of the manifold goals of Music Information Retrieval to provide new and intuitive ways to access music (e.g. to efficiently find music in online stores) and to automatically support the user in organizing his/her music collection. To this end, several techniques have been proposed. Although there exist many interesting approaches that are based on manually assigned meta-data (e.g. [21] or Musiclens[1]), we will solely concentrate on systems which rely on audio-based similarity calculations between music pieces. In general, such systems use the similarity information to automatically structure a music repository and aid the user in his/her exploration.

A very remarkable interface to discover new pieces and easily generate playlists is presented in [5]. From streams of music pieces (represented as discs) the user can simply pick out a piece to listen to or "collect" similar pieces by dragging a seed song into one of the streams. The different streams describe different moods. The number of released discs can be regulated for each mood separately by "tabs". Furthermore, the system invites users to experiment with playlists as all modifications can be undone easily by a so called *time-machine function*. Also combining playlists is facilitated through the intuitive drag-and-drop interface.

Other interfaces focus more on structuring and facilitating the access to existing collections instead of recommending new songs. Since in most cases, musical similarity is derived from a high-dimensional feature space, it is necessary to project the data into a lower-dimensional (latent) space in order to make it understandable to humans – a technique also commonly used in classical Information Retrieval [25]. For music, a frequently used approach is to apply *Self-Organizing Maps* (SOM) to arrange the collection on a 2-dimensional map that is intuitively readable by the user. We will explain the functionality of SOMs in Section 3.2. The first and most important approach that incorporated SOMs to structure music collections is Pampalk's *Islands of Music* interface [13, 16]. For the *Islands of Music*, a SOM is calculated on *Fluctuation Pattern* features (cf. Section 3.1.1). It visualizes the calculated SOM by applying a technique called *Smoothed Data Histogram* (cf. Section 3.3). Finally, a color model inspired by geographical maps is applied. Thus, on the resulting map, blue regions (oceans) indicate areas onto which very few pieces of music are mapped, whereas clusters containing a larger quantity of pieces are colored in brown and white (mountains and snow). In addition to this approach, several extensions have been proposed, e.g. the usage of Aligned SOMs [14] to enable a seamless shift of focus between different aspects of similarity. Furthermore, in [19] the interface has been extended by a hierarchical component to cope with very large music collections. In [12], SOMs are utilized for browsing in collections and intuitive playlist generation on portable devices. Other published approaches use SOM derivatives [11], similar techniques like FastMap [4], or graph-drawing algorithms to visualize the similarity of artists on portable devices [23]. The interface presented in [20] can utilize different approaches to map creation (including manual construction) and puts a focus on social interaction at playlist creation.

Another approach to assisting the user in browsing a music collection is spatialized music playback. In [22], an audio editor and browser is presented which makes use of the Princeton Scalable Display Wall with a 16-speaker surround system. In the so called *SoundSpace* browser, audio thumbnails of pieces close to the actual track are played simultaneously. In [3], sounds are represented as visual and sounding objects with specific properties. On a grid, the user can define a position from which all sounds that fall into a surrounding region ("aura") are played spatialized according to the position on the grid. [9] also deals with spatialized audio playback for usage in alternative music interfaces.

With our work, we primarily follow Pampalk's *Islands of Music* approach and (literally) raise it to the next dimension. Instead of just presenting a map, we generate a virtual landscape which encourages the user to freely navigate and explore the underlying music collection (cf. Figure 2). We also include spatialized audio playback. Hence, while moving through the landscape, the user hears audio thumbnails of close songs. Furthermore, we incorporate procedures from web-retrieval in conjunction with a SOM-labeling strategy to display words that describe the styles of music or images that are related to these styles in the different regions on the landscape.

3. TECHNICAL FUNDAMENTALS

In this section, we briefly introduce the underlying techniques of our interface. First, we describe the methods to compute similarities based on features extracted from the audio files. Second, we explain the functionality of the Self-Organizing Map which we use to cluster the high-dimensional data on a 2-dimensional map. Third, we review the smoothed data histogram approach, used to create a smooth terrain from a trained SOM. The last section concerns on the incorporated SOM-labeling strategy to display words from the web that describe the heard music. Since incorporation of related images is a straightforward extension of the presented procedures, details on this are given later in Section 4.3.4.

[1] `http://www.musiclens.de`

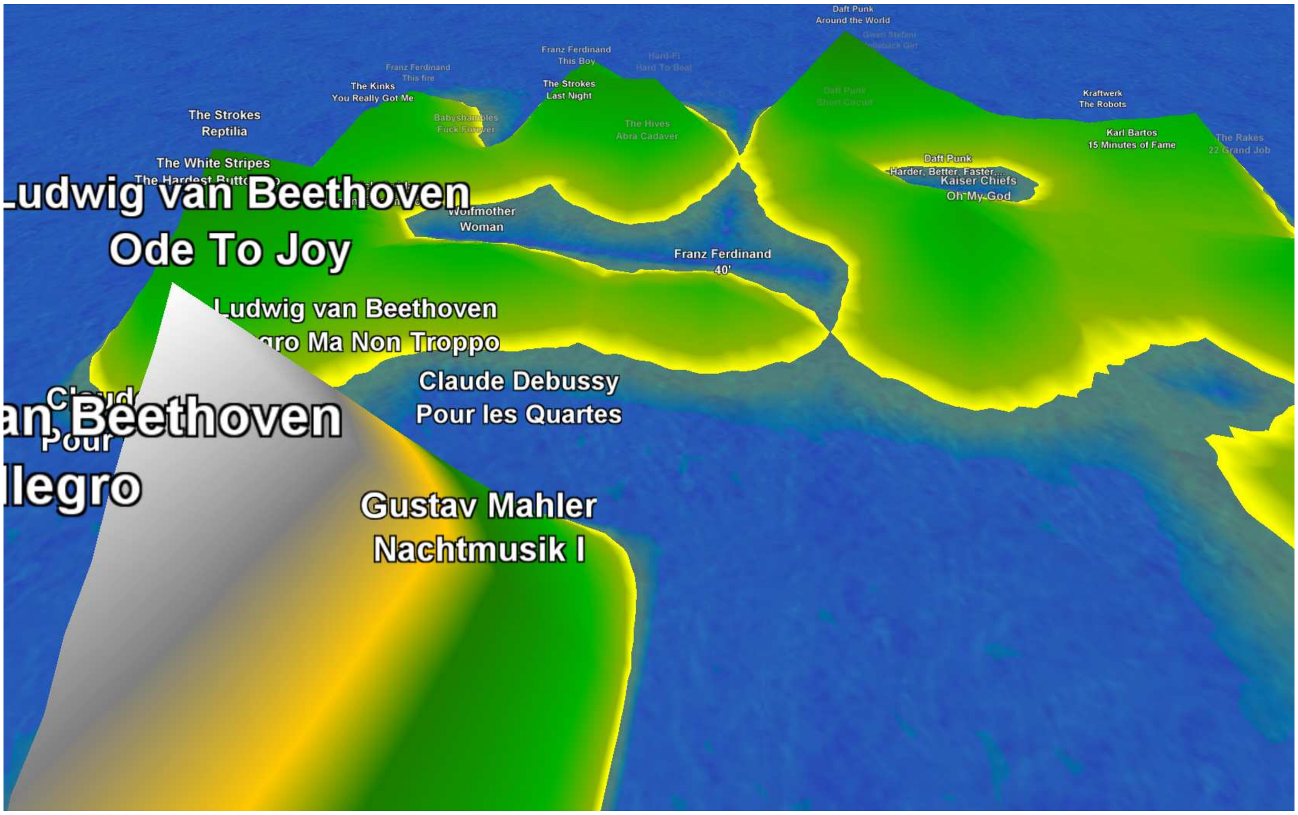

Figure 2: Screenshot of the user interface. The large peaky mountain in the front contains classical music. The classical pieces are clearly separated from the other musical styles on the landscape. The island in the left background contains Alternative Rock, while the islands on the right contain electronic music.

3.1 Audio-based Similarity

3.1.1 Fluctuation Patterns

The rhythm-based *Fluctuation Patterns* model the periodicity of the audio signal and were first presented in [13, 16]. In this section, we only sketch the main steps in the computation of these features. For more details, please consult the original sources. The feature extraction process is carried out on short segments of the signal, i.e. every third 6 second sequence. In a first step, a *Fast Fourier Transformation* (FFT) is applied to these audio segments. From the frequencies of the resulting spectrum, 20 critical-bands are calculated according to the bark scale. Furthermore, spectral masking effects are taken into account. In a next step, several loudness transformations are applied. As a consequence, the processed piece of music is represented by a number of feature matrices that contain information about the perceived loudness at a specific point in time in a specific critical-band. In the following stage another FFT is applied, which gives information about the amplitude modulation. These so-called fluctuations describe rhythmic properties by revealing how often a specific frequency reoccurs. Additionally, a psychoacoustic model of fluctuation strength is applied since the perception of the fluctuations depends on their periodicity, e.g. reoccurring beats at 4 Hz are discerned most intensely. In a final step, the median of all Fluctuation Pattern representations for the processed piece is calculated to obtain a unique, typically 1,200-dimensional feature vector for each piece of music. To use a set of such feature vectors for defining similarities between pieces, e.g. the Euclidean distances between the feature vectors must be calculated. For our purposes, we prefer to operate directly on the set of feature vectors since the quality of the resulting SOM is usually better when trained on feature data.

3.1.2 Alternative Similarity Measures

Although the following two similarity measures are not used in the current implementation, we briefly introduce them, since we plan to incorporate them (i.e. by combining them with a similarity matrix obtained via Fluctuation Patterns). First experiments yielded interesting and promising results. Both feature extraction algorithms are based on *Mel Frequency Cepstral Coefficients* (MFCCs). MFCCs give a coarse description of the envelope of the frequency spectrum and thus model timbral properties of a piece of music. Since MFCCs are calculated on time invariant frames of the audio signal, usually *Gaussian Mixture Models* (GMMs) are used to model the MFCC distributions of a whole piece of music. Similarity between two pieces of music A and B is then derived by drawing a sample from A's GMM and estimating the probability that this sample was created by B's GMM. The first MFCC-based similarity measure corresponds to the one described by Aucouturier et al. in [2]. The second measure has been proposed by Mandel and Ellis [10]. The measures basically differ in terms of the number and type of GMMs used and in calculation time.

3.2 The Self-Organizing Map

The SOM [7] is an unsupervised neural network that organizes multivariate data on a usually 2-dimensional map in such a manner that data items which are similar in the high-dimensional space are projected to similar locations on the map. Basically, the SOM consists of an ordered set of map units, each of which is assigned a "model vector" in the original data space. The set of all model vectors of a SOM is called its "codebook". There exist different strategies to initialize the codebook. We simply use a random initialization. For training, we use the batch SOM algorithm: In a first step, for each data item x, the Euclidean distance between x and each model vector is calculated. The map unit possessing the model vector that is closest to a data item x is referred to as "best matching unit" and is used to represent x on the map. In the second step, the codebook is updated by calculating weighted centroids of all data elements associated to the corresponding model vectors. This reduces the distances between the data items and the model vectors of the best matching units and also their surrounding units, which participate to a certain extent in the adaptations. The adaptation strength decreases gradually and depends on both distance of the units and iteration cycle. This supports the formation of large clusters in the beginning and a fine-tuning toward the end of the training. Usually, the iterative training is continued until a convergence criterion is fulfilled.

3.3 Smoothed Data Histogram

An approach that creates appealing visualizations of the data clusters of a SOM is the *Smoothed Data Histogram* (SDH), proposed in [17]. An SDH creates a smooth height profile (where height corresponds to the number of items in each region) by estimating the density of the data items over the map. To this end, each data item votes for a fixed number of best matching map units. The selected units are weighted according to the degree of the matching. The votes are accumulated in a matrix describing the distribution over the complete map. After each piece of music has voted, the resulting matrix is interpolated in order to obtain a smooth visualization. Additionally, a color map can be applied to the interpolated matrix to emphasize the resulting height profile. We apply a color map similar to the one used in the *Islands of Music*, to give the impression of an island-like terrain.

3.4 SOM-Labeling

An important aspect of our user interface is the incorporation of related information extracted automatically from the web. In particular, we intend to augment the landscape with music-specific terms that are commonly used to describe the music in the current region. We exploit the web's collective knowledge to figure out which words are typically used in the context of the represented artists. Details on the retrieval of these words are given in Section 4.3.

Once we have gathered a list of typical words for each artist, we are in need of both a strategy for transferring the list of artist-relevant words to the specific tracks on the landscape, as well as a strategy for determining those words that discriminate between the music in one region of the map and those in another (e.g. *music* is not a discriminating word, since it occurs very frequently for all artists). We decided to apply the SOM-labeling strategy proposed by Lagus and Kaski [8]. In their heuristically motivated weighting scheme, the relevance w_{tc} of a term for a cluster is calculated as

$$w_{tc} = (tf_{tc} / \sum_{t'} tf_{t'c}) \cdot \frac{(tf_{tc} / \sum_{t}' tf_{t'c})}{\sum_{c'} (tf_{tc'} / \sum_{t'} tf_{t'c'})}, \qquad (1)$$

where tf_{tc} denotes the frequency of term t in cluster c. We simply determine the term frequency for a term in each cluster as

$$tf_{tc} = \sum_{a} f_{ac} \cdot tf_{ta}, \qquad (2)$$

where f_{ac} gives the number of tracks of artist a in cluster c and tf_{ta} the term frequency of term t for artist a. For each cluster, we use the 8 highest weighted terms to describe its content.

We also experimented with the χ^2-test to find the most discriminating terms for each cluster. Usually, the χ^2-test is a well-applicable method to reduce the feature space in text categorization problems (see e.g. [26] for a detailed discussion). However, we found the Lagus and Kaski approach to yield better results for our task.

4. APPLICATION REALIZATION

In the following, we describe the realization of the user interface. First, the concept and the philosophy of the interface are explained. Second, we want to describe a typical use-case for the application. Third, we describe how we incorporate the techniques reviewed in Section 3 to create the application. Finally, we will make some remarks on the implementation.

4.1 Interface Concept

Our intention is to provide an interface to music collections detached from the conventional computer interaction metaphors. The first step toward this is the creation of an artificial but nevertheless appealing landscape that encourages the user to explore interactively.e Furthermore, we refrain from the usage of standard UI-components, contained in almost every window toolkit. Rather than constructing an interface that relies on the classical point-and-click scheme best controlled through a mouse, we designed the whole application to be controllable with a standard game pad as used for video game controlling. From our point of view, a game pad is perfectly suited for exploration of the landscape as it provides the necessary functionality to navigate in three dimensions whilst being easy to handle. However, we also included the option to navigate with a mouse in cases where no game pad is available (which has confirmed our opinion that a mouse is not the perfectly suited input device for this application). The controlling via a game pad also suggests a closeness to computer games which is absolutely intended since we aim at creating an interface that is fun to use. Therefore, we kept the controlling scheme very simple (cf. Figure 3).

Another important characteristic of the interface is the fact that the music surrounding the listener is played during navigation. Hence, it is not necessary to select each song manually and scan it for interesting parts. While the user explores the collection he/she is automatically presented with thumbnails from the closest music pieces, giving immediate auditory feedback on the style of music in the current region. Thus, the meaningfulness of the spatial distribution

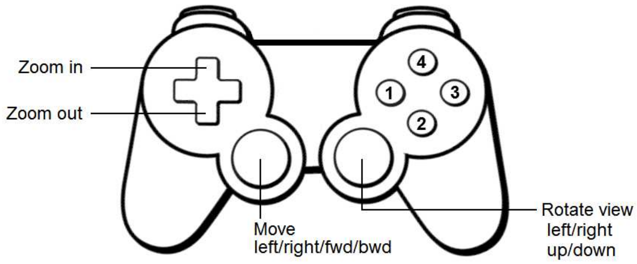

Figure 3: Controlling scheme of the application. For navigation, only the two analog sticks are necessary. The directional buttons up and down are used to arrange the viewer's distance to the landscape. The buttons 1-4 are used to switch between the different labeling modes. Mode (1) displays just the plain landscape without any labels. In mode (2), artist name and song name, as given by the id3 tags of the mp3s, are displayed (default). Mode (3) shows typical words that describe the heard music, while in mode (4), images from the web are presented that are related to the artists and the descriptions.

of music pieces in the virtual landscape can be experienced directly.

Finally, we aim at incorporating information beyond the pure audio signal. In human perception, music is always tied to personal and cultural influences that can not be captured by analyzing the audio. For example, cultural factors comprise time-dependent phenomena, marketing, or even influences by the peer group. Since we also intend to account for some of these aspects to provide a comprehensive interface to music collections, we exploit information available on the web. The web is the best available source for information regarding social factors as it represents current trends like no other medium.

Our interface provides four modes to explore the landscape. In the default mode, it displays the artist and track names as given by the id3 tags of the mp3 files. Alternatively, this information can be hidden, which focuses the exploration on the spatialized audio sensation. In the third mode, the landscape is enriched with words describing the heard music. The fourth mode displays images gathered automatically from the web that are related to the semantic descriptors and the contained artists, which further deepens the multimedia experience. Screenshots from all four modes can be seen in Figure 4.

In summary, we propose a multimedia application that examines several aspects of music and incorporates information on different levels of music perception - from the pure audio signal to culturally determined meta-descriptions. Thus, our application also offers the opportunity to discover new aspects of music. We think that this makes our new approach an interesting medium to explore music collections, unrestrained by stereotyped thinking.

4.2 The User's View

Currently, the application is designed to serve as an exhibit in a public space. Visitors are encouraged to bring their own collection, e.g. on a portable mp3 player and explore their collection through the landscape metaphor. Thus, the main focus was not on the applicability as a product ready to use at home. However, this could be achieved with little effort by incorporating standard music player functionalities.

In the application's current state, the process is invoked by the user through connecting his/her portable music player via an USB port. While the contained mp3 files are being analyzed, small, colored cubes pop up in the sky. The cubes display the number of items left to process. Thus, they serve as progress indicator. When the processing of an audio track is finished, the corresponding cube drops down and splashes into the sea. After all tracks have been processed, an island landscape that contains the tracks emerges from the sea. Then, it's the user's turn to explore the collection.

The three-dimensional landscape is projected onto the wall in front of the user. While moving through the terrain, the closest sounds with respect to the listener's current position can be heard from the directions where the pieces are located to emphasize the immersion. Thus, in addition to the visual grouping of pieces conveyed by the islands metaphor, islands are also perceived in an auditory manner, since one can hear typical sound characteristics for different regions. For optimal sensation of these effects, sounds are output via a 5.1 surround audio system.

Detaching the USB storage device (i.e. the mp3 player) causes all tracks on the landscape to immediately stop playback. The game pad is disabled and the viewer's position is moved back to the start. Subsequently, the landscape sinks back into the sea, giving the next user the opportunity to explore his/her collection.

4.3 The Engineer's View

4.3.1 *Audio Feature Extraction*

Our application automatically detects new storage devices on the computer and scans them for mp3 files. From the contained files, at most 50 are (randomly) chosen. We have limited the number of files to process mainly for time reasons, since the application should be accessible to many users. From the chosen audio files, the middle 30 seconds are extracted and analyzed. These 30 seconds also serve as looped audio thumbnail in the landscape. The idea is to extract the audio features (i.e. Fluctuation Patterns) only on a consistent and typical section of the track.

4.3.2 *Landscape Generation*

After training a SOM on the extracted audio features and computing an SDH, we need to create a three-dimensional landscape model that contains the musical pieces. However, in the SOM representation, the pieces are only assigned to a cluster rather than to a precise position. Thus, we have to elaborate a strategy to place the pieces on the landscape. The simplest approach would be to spread them randomly in the region of their corresponding map unit. This method has two drawbacks. The first is the overlap of labels, which occurs especially frequently for pieces with long names and results in cluttered maps. The second drawback is the loss of ordering of the pieces. It is desirable to have placements on the map that reflect the positions in feature space in some manner.

To address these problems, we decided to define a minimum distance d between the pieces that can be simply maintained by placing the pieces on circles around the map unit's center. To preserve at least some of the distance informa-

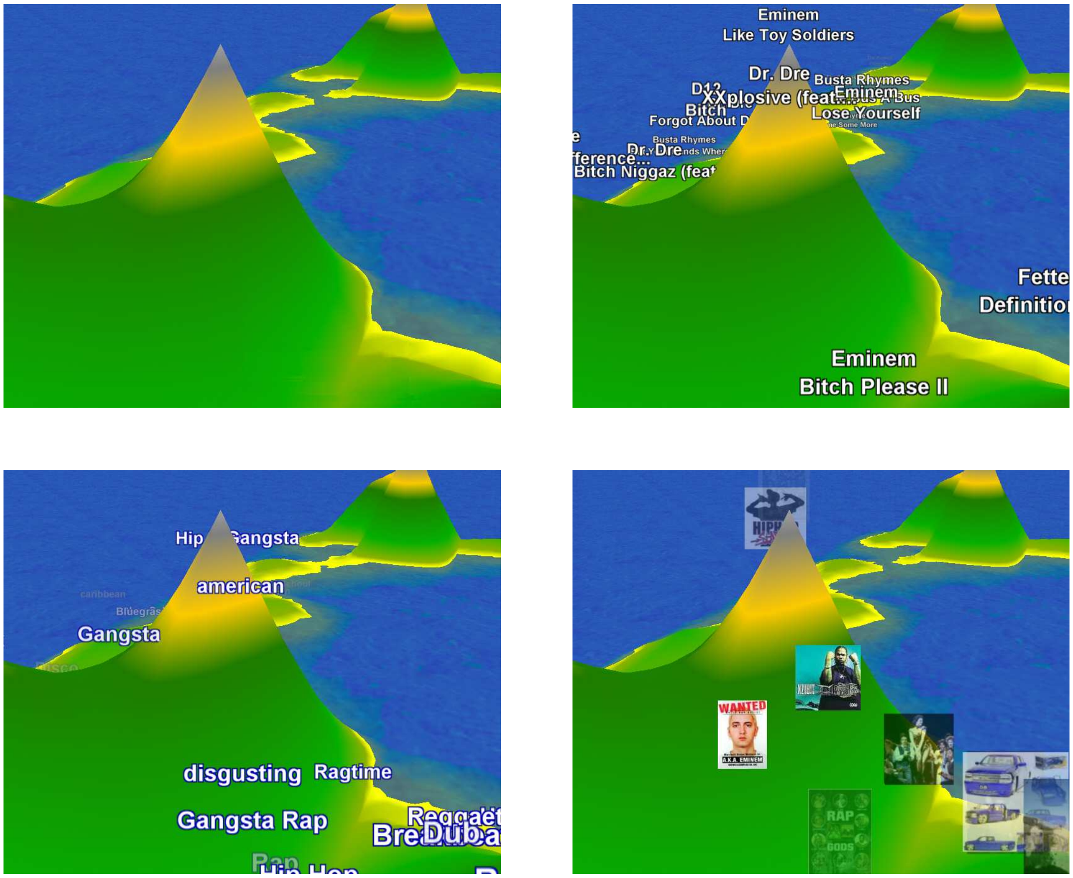

Figure 4: Four screenshots from the same scene in the four different modes. The upper left image depicts the plain landscape in mode 1. The image in the upper right shows mode 2, where artist and song name are displayed. Since this island contains Rap music, we find tracks of artists like *Eminem* and *Dr. Dre*. Mode 3 (lower left) shows typical words that describe the music, such as *Gangsta*, *Rap*, *Hip Hop*, or even *disgusting*. The lower right image depicts a screenshot in mode 4, where related images from the web are presented on the landscape. In this case, these images show the Rap artists *Eminem* and *Xzibit*, as well as a tour poster and pictures of "pimped" cars.

tion from feature space, we sort all pieces according to their distance to the model vector of their best matching unit in feature space. The first item is placed in the center of the map unit. Then, on the first surrounding circle (which has a radius of d, to meet the minimum distance), we can at most place the next $2\pi = 6$ pieces keeping distance d, since it has a perimeter of $2d\pi$. In the next circle (radius $2d$), we may place $4\pi = 12$ pieces and so on. These values are constant and can be calculated in advance. For map units with few items, we scale up the circle radii, to distribute the pieces as far as possible. As a result, the pieces most similar to the cluster centers are kept in the centers of their map units and also distances are preserved to some extent. However, this is a very heuristic approach that is far from perfect (for example, orientation, i.e. distances to other clusters, is currently ignored).

4.3.3 Term Retrieval

To be able to display semantic content on the landscape, i.e. words that describe the music, we have to extract specific types of information from the web. While it is difficult to find information specific to certain songs, it is feasible to extract information describing the general style of an artist. This information can be used to calculate artist similarity [24], perform artist to genre classification [6], or, most directly related to our task, help to organize and describe music collections [15]. In the cited references, this is realized by invoking Google with a query like `''artist name'' music review` and downloading the first 50 returned pages. For these pages term frequency (*tf*) and document frequency (*df*) are derived for either single words, bigrams, or trigrams and combined into the well known $tf \times idf$ measure. All these techniques have in common that, at least in the experimental settings, time was not a constraint. In contrast, we are very limited in time and resources as we have to extract the desired information while the audio files are being processed and the progress is visualized. Thus, instead of using the expensive data retrieval methods proposed in the papers mentioned above, i.e. retrieval of about 50 pages per artist, we simplify the search for musical style by formulating the query `''artist name'' music style`. Using this query, we retrieve Google's result page containing the first 100 pages. Instead of downloading each of the returned sites, we directly analyze the complete result page, i.e. the "snippets" presented. Thus, we can reduce the effort to just one web page per artist. To avoid the occurrence of totally unrelated words, we use a domain-specific dictionary, which is basically a shortened version of the dictionary used in [15]. After obtaining a term frequency representation of the dictionary vector for each artist, we determine the important words for each cluster as described in Section 3.4. The resulting labels are distributed randomly across the map unit.

4.3.4 Image Retrieval

To display images related to the artists and the describing words, we make use of the images search function of Yahoo!. We simply use the artist name or the term itself as query. Furthermore, we restrict results to images with dimensions in the range of 30 to 200 pixels. To find the three most important artists for each cluster, we basically perform the same ranking method as for the terms (see sections 4.3.3 and 3.4). For each important artist and every term, one of the first three images is chosen randomly and displayed on the map.

4.4 Implementation Remarks

The software is written exclusively in Java. For the realization of the three-dimensional landscape, we utilize the Xith3D scenegraph library[2], which runs on top of jogl and OpenGL. Spatialized surround sound is realized via Sound3D, joal[3], and OpenAL. To access the game controller we use the Joystick Driver for Java[4]. At the moment, the software runs on a Windows machine. Since all required libraries are also available for Linux, it is planned to port the software soon to this platform.

Since the processing of the songs is a very resource consuming but also very time critical task, we need a high-end PC to reduce the user's waiting time to a minimum. Thus, we rely on a dual core state-of-the-art-machine to quickly calculate the sound characteristics and download all web pages and images, as well as display the progress to the user without latencies.

5. QUALITATIVE EVALUATION

We conducted a small user study to gain insights into the usability of the application. Therefore, we asked 8 participants to tell us their impressions after using the interface. In general, responses were very positive. People were impressed by the possibility to explore and listen to a music collection by cruising through a landscape. While the option to display related images on the landscape has been considered mainly as a nice gimmick, the option to display related words was rated as a valuable add-on, even if some of the displayed words were confusing for some users. The controlling by gamepad was intuitive for all users.

Sceptical feedback was mainly caused by music auralization in areas where different styles collide. However, in general, auralization was considered positive, especially in regions containing Electronic Dance Music, Rap/HipHop, or Classical Music, since it assists in quickly uncovering groups of tracks from the same musical style. Two users suggested to create larger landscapes to allow focused listening to certain tracks in crowded regions.

[2]http://www.xith3d.org

[3]https://joal.dev.java.net

[4]http://sourceforge.net/projects/javajoystick

6. DISCUSSION AND FUTURE WORK

We have presented an innovative approach to accessing music collections. Using our virtual reality, game-like interface, it is possible to explore the contents in a playful manner. Furthermore, we have modified existing web retrieval approaches to enrich the generated landscape with semantic information related to the music.

In it's current state, the application has a focus on interactive exploration rather than on providing full functionality to replace existing music players. However, we can easily extend the application to provide such useful methods as automatic playlist generation. To this end, we can give the user the option to determine a start and an end song on the map. Given this information, we can then find a path along the distributed pieces on the map. Furthermore, we can easily visualize such paths and provide some sort of "autopilot mode", where the movement through the landscape is done automatically by following the playlist path. One of the central question that arises is how to explicitly select specific tracks in the landscape. At the moment, all pieces in the surrounding region are played for auditory exploration, but there is no possibility to focus exclusively on one track. We are currently exploring three different options. The first would be to provide a cross-hair that can be controlled by the directional buttons of the game pad. The second option would be to reserve one (or two) buttons to scan through all, or at least the closest tracks that are visible. In both cases, selection of the track would need an additional button to confirm the selection. The third option would display a number next to the four closest pieces and utilize the buttons 1–4 (cf. Figure 3) to directly select one of these tracks. Before making a definitive choice, we will have to carry out further user experiments and gain more experience in practical scenarios. With the ability to select specific tracks, we could introduce focused listening and also present additional track-specific meta-data for the currently selected track. For example, we could display further id3 tags like album or track length, as well as lyrics or album covers. In future work, we will also address the problem of visualizing very large collections. Currently, we have limited the number of pieces to 50 for time reasons and for reasons of clarity. An option would be to incorporate hierarchical extensions as proposed in [19].

Another possible extension of the application concerns force feedback. As many game pads have built-in force feedback functionality, it would be an interesting option to involve an additional human sense, namely the tactile perception. First experiments regarding exploration of music collections based on tactile feedback have been made in [18, 1]. In our case, the primary goal would not be to develop a tactile description for musical pieces, but simply to deepen the immersion in specific regions, e.g. regions that contain many pieces with very strong beats.

7. ACKNOWLEDGMENTS

This research is supported by the Austrian Fonds zur Förderung der Wissenschaftlichen Forschung (FWF) under project number L112-N04 and by the Vienna Science and Technology Fund (WWTF) under project number CI010 (Interfaces to Music). The Austrian Research Institute for Artificial Intelligence acknowledges financial support by the Austrian ministries BMBWK and BMVIT.

Special thanks are due to the students who implemented vital parts of the project, especially Richard Vogl, who designed the first interface prototype and Klaus Seyerlehner, who implemented high-level feature extractors.

8. REFERENCES

[1] M. Allen, J. Gluck, K. MacLean, and E. Tang. An initial usability assessment for symbolic haptic rendering of music parameters. In *ICMI '05: Proc. of the 7th international conference on Multimodal interfaces*, New York, NY, USA, 2005. ACM Press.

[2] J.-J. Aucouturier, F. Pachet, and M. Sandler. "The Way It Sounds": Timbre Models for Analysis and Retrieval of Music Signals. *IEEE Transactions on Multimedia*, 7(6):1028–1035, December 2005.

[3] E. Brazil and M. Fernström. Audio information browsing with the sonic browser. In *Coordinated and Multiple Views In Exploratory Visualization (CMV03)*, London, UK, 2003.

[4] P. Cano, M. Kaltenbrunner, F. Gouyon, and E. Batlle. On the Use of Fastmap for Audio Retrieval and Browsing. In *Proc. of the International Conference on Music Information Retrieval (ISMIR'02)*, Paris, France, 2002.

[5] M. Goto and T. Goto. Musicream: New Music Playback Interface for Streaming, Sticking, and Recalling Musical Pieces. In *Proc. of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, London, UK, 2005.

[6] P. Knees, E. Pampalk, and G. Widmer. Artist Classification with Web-based Data. In *Proc. of 5th International Conference on Music Information Retrieval (ISMIR'04)*, Barcelona, Spain, October 2004.

[7] T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, 3rd edition, 2001.

[8] K. Lagus and S. Kaski. Keyword selection method for characterizing text document maps. In *Proc. of 9th International Conference on Artificial Neural Networks (ICANN'99)*, volume 1, London, 1999. IEEE.

[9] D. Lübbers. Sonixplorer: Combining Visualization and Auralization for Content-based Exploration of Music Collections. In *Proc. of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, London, UK, 2005.

[10] M. Mandel and D. Ellis. Song-Level Features and Support Vector Machines for Music Classification. In *Proc. of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, London, UK, 2005.

[11] F. Mörchen, A. Ultsch, M. Nöcker, and C. Stamm. Databionic visualization of music collections according to perceptual distance. In *Proc. of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, London, UK, 2005.

[12] R. Neumayer, M. Dittenbach, and A. Rauber. PlaySOM and PocketSOMPlayer, Alternative Interfaces to Large Music Collections. In *Proc. of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, London, UK, 2005.

[13] E. Pampalk. Islands of Music: Analysis, Organization, and Visualization of Music Archives. Master's thesis, Vienna University of Technology, 2001.

[14] E. Pampalk, S. Dixon, and G. Widmer. Exploring music collections by browsing different views. *Computer Music Journal*, 28(2):49–62, 2004.

[15] E. Pampalk, A. Flexer, and G. Widmer. Hierarchical organization and description of music collections at the artist level. In *Proc. of the 9th European Conference on Research and Advanced Technology for Digital Libraries (ECDL'05)*, Vienna, Austria, 2005.

[16] E. Pampalk, A. Rauber, and D. Merkl. Content-based organization and visualization of music archives. In *Proc. of the ACM Multimedia*, Juan les Pins, France, December 1-6 2002. ACM.

[17] E. Pampalk, A. Rauber, and D. Merkl. Using smoothed data histograms for cluster visualization in self-organizing maps. In *Proc. of the International Conference on Artifical Neural Networks (ICANN'02)*, Madrid, Spain, 2002.

[18] S. Pauws, D. Bouwhuis, and B. Eggen. Programming and enjoying music with your eyes closed. In *CHI '00: Proc. of the SIGCHI conference on Human factors in computing systems*, New York, NY, USA, 2000. ACM Press.

[19] M. Schedl. An explorative, hierarchical user interface to structured music repositories. Master's thesis, Vienna University of Technology, December 2003.

[20] I. Stavness, J. Gluck, L. Vilhan, and S. Fels. The MUSICtable: A Map-Based Ubiquitous System for Social Interaction with a Digital Music Collection. In *Proc. of the 4th International Conference on Entertainment Computing (ICEC 2005)*, Sanda, Japan, 2005.

[21] M. Torrens, P. Hertzog, and J.-L. Arcos. Visualizing and Exploring Personal Music Libraries. In *Proc. of 5th International Conference on Music Information Retrieval (ISMIR'04)*, Barcelona, Spain, 2004.

[22] G. Tzanetakis and P. Cook. Marsyas3D: A Prototype Audio Browser-Editor Using a Large Scale Immersive Visual Audio Display. In *Proc. of the International Conference on Auditory Display*, 2001.

[23] R. van Gulik, F. Vignoli, and H. van de Wetering. Mapping music in the palm of your hand, explore and discover your collection. In *Proc. of 5th International Conference on Music Information Retrieval (ISMIR'04)*, Barcelona, Spain, 2004.

[24] B. Whitman and S. Lawrence. Inferring descriptions and similarity for music from community metadata. In *Proc. of the 2002 International Computer Music Conference*, Goteborg, Sweden, September 2002.

[25] J. A. Wise, J. J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow. Visualizing the Non-Visual: Spatial analysis and interaction with information from text documents. In *Proc. of the 1995 IEEE Symposium on Information Visualization (INFOVIS'95)*, Atlanta, Georgia, 1995.

[26] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In D. H. Fisher, editor, *Proc. of ICML-97, 14th International Conference on Machine Learning*, Nashville, US, 1997.