

Exploring Music Collections in Virtual Landscapes

Peter Knees, Markus Schedl, Tim Pohle,
and Gerhard Widmer
Johannes Kepler University Linz

A user interface to music repositories called nepTune creates a virtual landscape for an arbitrary collection of digital music files, letting users freely navigate the collection. Automatically extracting features from the audio signal and clustering the music pieces accomplish this. The clustering helps generate a 3D island landscape.

The ubiquity of digital music is 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, we can argue that recent technical advancements in audio coding and the associated enormous success of portable MP3 players, especially Apple's iPod, have immensely added to forming the zeitgeist.

These developments are profoundly changing the way people use music. Music is becoming a commodity that is traded electronically, exchanged, shared (legally or not), and even used as a means for social communication and display of personality (witness the huge number of people putting their favorite tracks on their personal sites, such as MySpace).

Despite these rapid changes in the way people use music, methods of organizing music collections on computers and music players has basically remained the same. Owners of digital music collections traditionally organize their thousands of audio tracks in hierarchical directories, often structured according to the common scheme: genre then artist then album then track. Indeed, people don't have much of a choice, given the options offered by current music players and computers.

The rapidly growing research field of music information retrieval is developing the technological foundations for a new generation of more intelligent music devices and services. Researchers

are creating algorithms for audio and music analysis, studying methods for retrieving music-related information from the Internet, and investigating scenarios for using music-related information for novel types of computer-based music services. The range of applications for such technologies is broad—from automatic music recommendation services through personalized, adaptive radio stations, to novel types of intelligent, reactive musical devices and environments.

At our institute, we are exploring ways of providing new views on the contents of digital music collections and new metaphors for interacting with music and collections of music pieces. At the confluence of artificial intelligence, machine learning, signal processing, data and Web mining, and multimedia, we develop algorithms for analyzing, interpreting, and displaying music in ways that are interesting, intuitive, and useful to human users. In this article, we describe a particular outcome of this research—an interactive, multimodal interface to music collections, which we call nepTune.

Our approach

The general philosophy underlying nepTune is that music collections should be structured (automatically, by the computer) and presented according to intuitive musical criteria. In addition, music interfaces should permit and encourage the creative exploration of music repositories and new ways of discovering hidden treasures in large collections. To make this kind of philosophy more popular, our first application is an interactive exhibit in a modern science museum.

The nepTune interface offers an original opportunity to playfully explore music by creating an immersive virtual reality founded in the sounds of a user's digital audio collection. We want the interface to be fun to use and engage people. The basic ingredients are as follows: Using intelligent audio analysis, nepTune clusters the pieces of music according to sound similarity. Based on this clustering, the system creates a 3D island landscape containing the pieces (see Figure 1). Hence, the resulting landscape groups similar-sounding pieces together. The more similar pieces the user owns, the higher the terrain in the corresponding region. The user can move through the virtual landscape and explore his or her collection. This visual approach essentially follows the islands of music metaphor (see Figure 2).¹ Each music collection created has its own unique characteristics and landscape. In addition

to seeing the music pieces in the landscape, the listener hears the pieces closest to his or her current position. Thus, the user gets an auditory impression of the musical style in the surrounding region, via a 5.1 surround sound system.

Furthermore, listeners can enrich the landscape with semantic and visual information acquired via Web retrieval techniques. Instead of displaying the song title and performing artist on the landscape, the user can choose to see words that describe or images related to the heard music. Thus, besides a purely audio-based structuring, nepTune also offers more contextual information that might trigger new associations in the listener/viewer, thus making the experience more interesting and rewarding.

Application realization

Here, we describe the realization of the nepTune music interface.

Interface concept

Our intention is to provide an interface to music collections that goes beyond conventional computer interaction metaphors. The first step toward this is to create an artificial but nevertheless appealing landscape that encourages the user to explore a music collection interactively. Furthermore, we refrain from using the kind of standard user interface 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 made the whole application controllable with a standard game pad such as those used for video games. 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 3D while being easy to handle. Furthermore, the resemblance to computer games is absolutely intentional. Therefore, we kept the controlling scheme simple (see Figure 3).

As mentioned before, another important interface characteristic is that it plays the music surrounding the listener during navigation. Hence, it's not necessary to select each song manually and scan it for interesting parts. While the user explores the collection, he or she automatically hears audio thumbnails from the closest music pieces, giving immediate auditory feedback on the style of music in the current region. Thus, users directly experience the meaningfulness of the spatial distribution of music pieces in the virtual landscape.

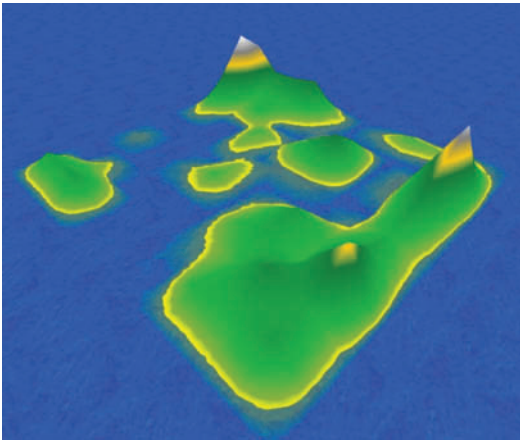


Figure 1. An island landscape created from a music collection. Listeners explore the collection by freely navigating through the landscape and hearing the music typical for the region around his or her current position.

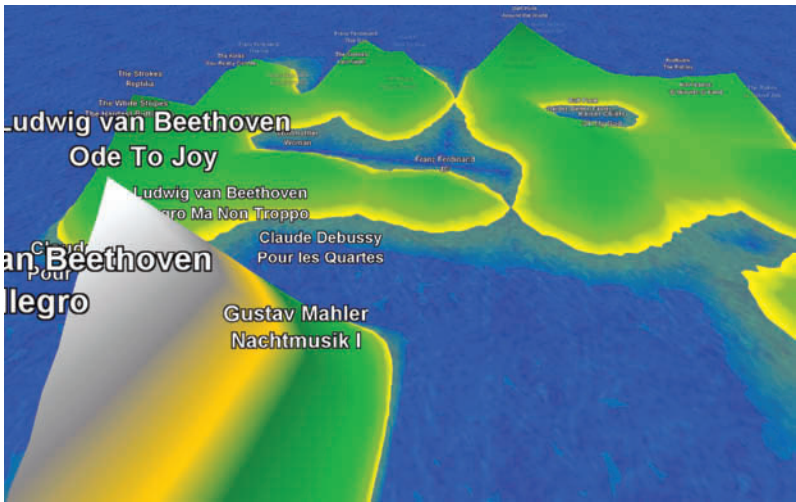


Figure 2. Screen shot of the nepTune interface. The large peaky mountain in the front contains classical music. The classical pieces are clearly separated from the other musical styles on the landscape. The island in the left background contains alternative rock, while the islands on the right contain electronic music.

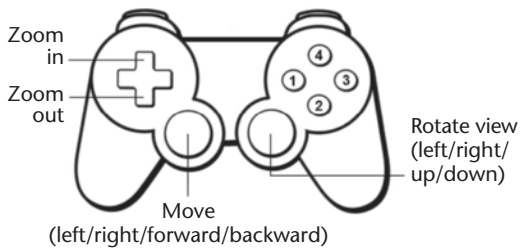


Figure 3. The controlling scheme of nepTune. For navigation, only the two analog sticks are necessary. The directional buttons up and down arrange the viewer's distance to the landscape. Buttons 1 through 4 switch between the different labeling modes.

Finally, we want to incorporate information beyond the pure audio signal. In human perception, music is always tied to personal and cultural influences that analyzing the audio can't capture. For example, cultural factors comprise time-dependent phenomena, marketing, or even influences by the user's peer group. Since we also intend to account for some of these aspects to

Background

At the heart of many intelligent digital music applications are the notion of musical similarity and computational methods for estimating the similarity of music pieces—as it might be perceived by human listeners. The most interesting, but also most challenging, approach to accomplish this is to infer the similar information directly from the audio signal via relevant feature extraction. Music similarity measures have become a large research topic in music information retrieval; an introduction to this topic is available elsewhere.¹⁻³

The second major step in creating a nepTune-like interface is the automatic structuring of a collection, given pairwise similarity relations between the individual tracks. This is essentially an optimization problem: to place objects into a presentation space so that pairwise similarity relations are preserved as much as possible. In the field of music information retrieval, a frequently used approach is to apply a self-organizing map (SOM) to arrange a music collection on a 2D map that the user can intuitively read.⁴

The most important approach that uses SOMs to structure music collections is the islands of music interface.⁵ The islands of music approach calculates a SOM on so-called fluctuation pattern features that model the music's rhythmic aspects. Applying a smoothed data histogram technique visualizes the calculated SOM. Finally, the system applies a color model inspired by geographical maps. Thus, on the resulting map, blue regions (oceans) indicate areas onto which few pieces of music are mapped, whereas clusters containing a larger quantity of pieces are colored in brown and white (mountains and snow). Several extensions have been proposed; for example, a hierarchical component to cope with large music collections.⁶ Similar interfaces use SOM derivatives⁷ or use SOMs for intuitive playlist generation on portable devices.⁸

With our work, we follow Pampalk's islands of music approach and (literally) raise it to the next dimension by providing an interactive 3D interface. Instead of just presenting a map, we generate a virtual landscape that encourages the user to freely navigate and explore the underlying music collection.

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 related to these styles in the different regions on the landscape.

References

1. E. Pampalk, S. Dixon, and G. Widmer, "On the Evaluation of Perceptual Similarity Measures for Music," *Proc. 6th Int'l Conf. Digital Audio Effects (DAFx)*, 2003, pp. 6-12; <http://www.elec.qmul.ac.uk/dafx03/proceedings/pdfs/dafx02.pdf>.
2. E. Pampalk, "Computational Models of Music Similarity and their Application to Music Information Retrieval," doctoral dissertation, Vienna Univ. of Technology, 2006.
3. T. Pohle, "Extraction of Audio Descriptors and their Evaluation in Music Classification Tasks," master's thesis, TU Kaiserslautern, German Research Center for Artificial Intelligence (DFKI), Austrian Research Inst. for Artificial Intelligence (OFAI), 2005; <http://kluedo.ub.uni-kl.de/volltexte/2005/1881/>.
4. T. Kohonen, *Self-Organizing Maps*, vol. 30, 3rd ed., Springer, Series in Information Sciences, Springer, 2001.
5. E. Pampalk, "Islands of Music: Analysis, Organization, and Visualization of Music Archives," master's thesis, Vienna Univ. of Technology, 2001.
6. M. Schedl, "An Explorative, Hierarchical User Interface to Structured Music Repositories," master's thesis, Vienna Univ. of Technology, 2003.
7. F. Mörchén et al., "Databionic Visualization of Music Collections According to Perceptual Distance," *Proc. 6th Int'l Conf. Music Information Retrieval (ISMIR)*, 2005, pp. 396-403; <http://ismir2005.ismir.net/proceedings/1051.pdf>.
8. R. Neumayer, M. Dittenbach, and A. Rauber, "PlaySOM and PocketSOMPlayer, Alternative Interfaces to Large Music Collections," *Proc. 6th Int'l Conf. Music Information Retrieval (ISMIR)*, 2005, pp. 618-623; <http://ismir2005.ismir.net/proceedings/1096.pdf>.

provide a comprehensive interface to music collections, we try to exploit information from the Web. The Web is the best available source for information regarding social factors as it represents current trends like no other medium. The method we propose next is but a first simple step toward capturing such aspects. More specialized Web-mining methods will be necessary for getting at truly cultural and social information.

nepTune provides four modes to explore the landscape. In the default mode, it displays the artist and track names as given by the MP3 files' ID3 tags (see <http://www.id3.org>). Alternatively, the system can hide this information, which

focuses the users' 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. Figure 4 shows screen shots from all four modes.

In summary, the nepTune multimedia application examines several aspects of music and incorporates information at different levels of music perception—from the pure audio signal to culturally determined metadescriptions—which

offers the opportunity to discover new aspects of music. This should make nepTune an interesting medium to explore music collections, unrestrained by stereotyped thinking.

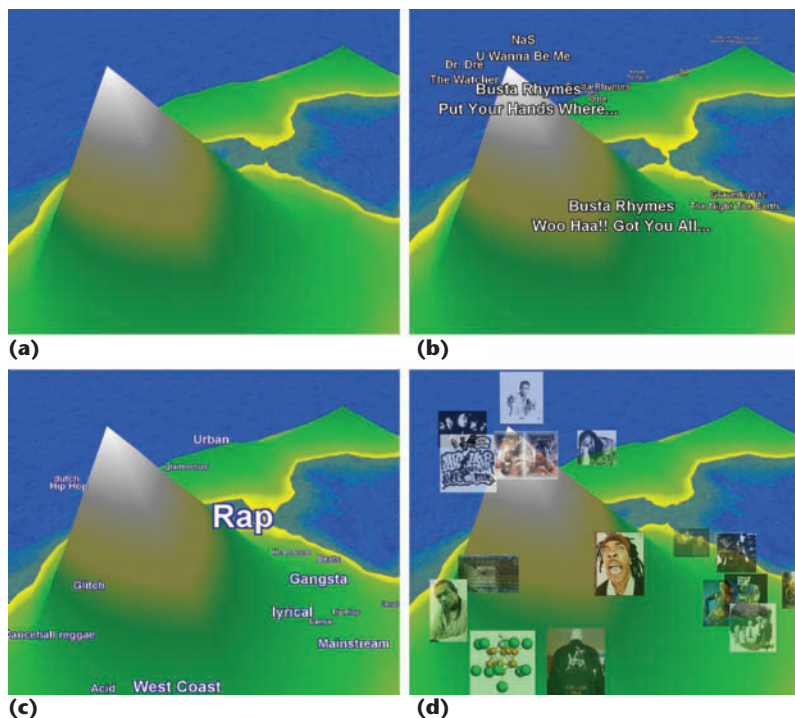
The user's view

We designed the current application to serve as an exhibit in a public space, that is, in a modern science museum. Visitors are encouraged to bring their own collection—for example, on a portable MP3 player—to explore their collection in the virtual landscape. Thus, our main focus was not on the system's applicability as a product ready to use at home. However, we could achieve this with little effort by incorporating standard music player functionalities.

In the application's current state, the user invokes the exploration process through connecting his or her portable music player via a USB port. nepTune automatically recognizes this, and the system then randomly extracts a predefined number of audio files from the player and starts to extract audio features (mel frequency cepstral coefficients, or MFCCs) from these. A special challenge for applications presented in a public space is to perform computationally expensive tasks, such as audio feature analysis, while keeping visitors motivated and convincing them that there is actually something happening. We decided to visualize the progress of audio analysis via an animation: small, colored cubes display the number of items left to process. For each track, a cube with the number of the track pops up in the sky. When an audio track's processing is finished, the corresponding cube drops down and splashes into the sea. After the system processes all tracks, an island landscape that contains the tracks emerges from the sea. After this, the user can explore the collection.

The system projects a 3D landscape onto the wall in front of the user. While moving through the terrain, the listener can hear the closest sounds with respect to his or her position from the directions of the music piece locations, to emphasize the immersion. Thus, in addition to the visual grouping of pieces conveyed by the islands metaphor, users can also perceive islands in an auditory manner, since they can hear typical sound characteristics for different regions. To provide optimal sensation related to these effects, the system outputs sounds via a 5.1 surround audio system.

Detaching the USB storage device (that is, the MP3 player) causes all tracks on the landscape to



immediately stop playback. This action also disables the game pad and moves the viewer's position back to the start. Subsequently, the landscape sinks back into the sea, giving the next user the opportunity to explore his or her collection.

Technical realization

Here, we explain the techniques behind nepTune: feature extraction from the audio signal, music piece clustering and projection to a map, landscape creation, and landscape enrichment with descriptive terms and related images.

Audio feature extraction. Our application automatically detects new storage devices on the computer and scans them for MP3 files. nepTune randomly chooses a maximum of 50 from the contained files. We have limited the number of files mainly for time reasons, helping make the application accessible to many users. From the chosen audio files, the system extracts and analyzes the middle 30 seconds. These 30 seconds also serve as looped audio thumbnails in the landscape. The idea is to extract the audio features only from a consistent and typical section of the track.

For calculating the audio features, we build upon the method proposed by Mandel and Ellis.² Like the foregoing approach by Aucouturier, Pachet, and Sandler,³ this approach is based on MFCCs, which model timbral properties. For

Figure 4. Screen shots from the same scene in the four different modes: (a) the plain landscape in mode 1; (b) mode 2, which displays artist and song name; (c) mode 3 shows typical words that describe the music, such as rap, gangsta, west coast, lyrical, or mainstream; (d) a screen shot in mode 4, where related images from the Web are presented on the landscape. In this case, these images show rap artists as well as related artwork.

We apply a color map similar to the one used in the islands of music, to give the impression of an island-like terrain.

each audio track, the system computes MFCCs on short-time audio segments (frames) to get a coarse description of the envelope of the individual analysis frames' frequency spectrum. The system then models the MFCC distribution over all of a track's frames via a Gaussian distribution with a full covariance matrix. Each music piece is thus represented by a distribution. The approach then derives similarity between two music pieces by calculating a modified Kullback-Leibler distance on the means and covariance matrices. Pairwise comparison of all pieces results in a similarity matrix, which is used to cluster similar pieces.

Landscape generation. To generate a landscape from the derived similarity information, we use a self-organizing map.⁴ The SOM organizes multivariate data on a usually 2D map in such a manner that similar data items in the high-dimensional space are projected to similar map locations. Basically, the SOM consists of an ordered set of map units, each of which is assigned a model vector in the original data space. A SOM's set of all model vectors is called its *codebook*. There exist different strategies to initialize the codebook; we use linear initialization.⁴ For training, we use the batch SOM algorithm: first, for each data item x , we calculate the Euclidean distance between x and each model vector.⁵ The map unit possessing the model vector closest to a data item x is referred to as the *best matching unit* and represents x on the map. In the second step, the codebook is updated by calculating weighted centroids of all data elements associated with the corresponding model vectors. This reduces the distances between the data items and the model vectors of the best matching units and their surrounding units, which participate to a certain extent in the adaptations. The adaptation strength decreases gradually and depends on

both unit distance and iteration cycle. This supports large cluster formation in the beginning of and fine tuning toward the end of the training. Usually, the iterative training continues until a convergence criterion is fulfilled.

To create appealing visualizations of the SOM's data clusters, we calculate a smoothed data histogram.⁶ An SDH creates a smooth height profile (where height corresponds to the number of items in each region) by estimating the data item density over the map. To this end, each data item votes for a fixed number n of best matching map units. The best matching unit receives n points, the second best $n - 1$, and so on. Accumulating the votes results in a matrix describing the distribution over the complete map. After each piece of music has voted, interpolating the resulting matrix yields a smooth visualization. Additionally, the user can apply a color map 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.

Based on the calculated SDH, we create a 3D landscape model that contains the musical pieces. However, the SOM representation only assigns the pieces 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. That has two drawbacks. The first is the overlap of labels, which occurs particularly often for pieces with long names and results in cluttered maps. The second drawback is the loss of ordering of the pieces. It's desirable to have placements on the map that reflect the positions in feature space in some way.

The solution we adopted is to define a minimum distance d between the pieces and place the pieces on concentric circles around the map unit's center so that this distance is always guaranteed. To preserve at least some of the similarity information 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), we can place at most ($2\pi \approx 6$) so that d is maintained (because the circle has a perimeter of $2d\pi$). The next circle (radius $2d$) can host up to ($4\pi \approx 12$) pieces, and so on. For map units with few items, we scale up the circle radii to distribute the pieces as far as possible

within the unit's boundaries. As a result, the pieces most similar to the cluster centers stay in the centers of their map units and distances are preserved to some extent. More complex (and computationally demanding) strategies are conceivable, but this simple approach works well enough for our scenario.

Displaying labels and images. An important aspect of our user interface is the incorporation of related information extracted automatically from the Web. The idea is to augment the landscape with music-specific terms 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.

To determine descriptive terms, we use a music description map.⁷ For each contained artist, we send a query consisting of the artist's name and the additional constraints "music style" to Google. We retrieve Google's result page containing links to the first 100 pages. Instead of downloading each of the returned sites, we directly analyze the complete result page—that is, the text snippets presented. Thus, we just have to download one Web page per artist. To avoid the occurrence of unrelated words, we use a domain-specific vocabulary containing 945 terms. Besides some adjectives related to moods and geographical names, these terms consist mainly of genre names, musical styles, and musical instrument types. For each artist, we count how often the terms from the vocabulary occur on the corresponding Web page (term frequency), which results in a term frequency vector.

After obtaining a vector for each artist, we need a strategy for transferring the list of artist-relevant words to specific points on the landscape and for determining those words that discriminate between the music in one region of the map and the music in other regions—for example, "music" is not a discriminating word, since it occurs frequently for all artists. For each unit, we sum up the term frequency vectors of the artists associated with pieces represented by the unit. The result is a frequency vector for each unit.

Using these vectors, we want to find the most descriptive terms for the units. We decided to apply the SOM labeling strategy proposed by Lagus and Kaski.⁸ Their heuristically motivated scheme exploits knowledge of the SOM's structure to enforce the emergence of areas with coherent descriptions. To this end, the approach

accumulates term vectors from directly neighboring units and ignores term vectors from a more distant neutral zone. We calculate the goodness score $G2$ of a term t as a descriptor for unit u as follows:

$$G2(t, u) = \frac{\left[\sum_{k \in A_0^u} F(t, k) \right]^2}{\sum_{i \in A_1^u} F(t, i)}$$

where $k \in A_0^u$ if the (Manhattan) distance of units u and k on the map is below a threshold r_0 , and $i \in A_1^u$ if the distance of u and i is greater than r_0 and smaller than some r_1 (in our experiments we set $r_0 = 1$ and $r_1 = 2$). $F(t, u)$ denotes the relative frequency of term t on unit u and is calculated as

$$F(t, u) = \frac{\sum_a f(a, u) \cdot tf(t, a)}{\sum_v \sum_a f(a, u) \cdot tf(v, a)}$$

where $f(a, u)$ gives artist a 's number of tracks on unit u and $tf(t, a)$ gives artist a 's term frequency of term t . Because many neighboring units contain similar descriptions, we try to find coherent parts of the music description map and join them to single clusters. The system then displays in the center of the cluster the most important terms for each cluster; it then randomly distributes the remaining labels across the cluster. The display size of a term corresponds to its score, $G2$.

To display images related to the artists and the describing words, we use Google's image search function. For each track, we include an image of the corresponding artist. For each term, we simply use the term itself as a query and randomly select one of the first 10 displayed images.

Implementation remarks

We exclusively wrote the software in Java. We implemented most of the application functionality in our Collection of Music Information Retrieval and Visualization Applications (CoMIRVA) framework, which is published under the GNU General Public License and can be downloaded from <http://www.cp.jku.at/comirva>. For the realization of the 3D landscape, we use the Xith3D scene graph library (see <http://xith.org>), which runs on top of Java OpenGL. Spatialized surround sound is realized via Sound3D and Java bindings for OpenAL (see <https://joal.dev.java.net>). To access the game controller we use the Joystick Driver for Java (see <http://sourceforge.net/projects/javajoystick>).



Figure 5. A mobile device running the prototype of the nepTune mobile version. (We postprocessed the display for better visibility.)

Currently, the software runs on a Windows machine. Because all required libraries are also available for Linux, we plan to port the software to that platform soon.

Qualitative evaluation

We conducted a small user study to gain insights into nepTune's usability. We asked eight people to play with the interface and tell us their impressions. In general, responses were positive. People reported that they enjoyed exploring and listening to a music collection by cruising through a landscape. While many considered the option of displaying related images on the landscape mainly a nice gimmick, many rated the option to display related words as a valuable add-on, even if some of the displayed words were confusing for some users. All users found controlling the application with a gamepad intuitive.

Skeptical feedback was mainly caused by music auralization in areas where different styles collide. However, in general, people rated auralization as positive, especially in regions containing electronic dance music, rap and hip-hop, or classical music, because it assists in quickly identifying groups of tracks from the same musical style. Two users suggested creating larger landscapes to allow more focused listening to certain tracks in crowded regions.

Future directions

In its current state, nepTune focuses on inter-

active 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. For example, we could let the user 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 occurs automatically by following the playlist path. By allowing the user to select specific tracks, we could also introduce focused listening and present additional track-specific metadata for the currently selected track. As in other music player applications, we could display further ID3 tags like album or track length, as well as lyrics or album covers.

Large collections (containing tens of thousands of tracks) present the biggest challenges. One option would be to incorporate a level-of-detail extension that uses the music descriptors extracted from the Web. At the top-most level, that is, the highest elevation, only broad descriptors like musical styles would be displayed. Reducing the altitude would switch to the next level of detail, making more distinct descriptors appear, along with important artists for that specific region. Single tracks could then be found at the most detailed level. This would emphasize the relatedness of the interface to geographical maps, and the application would act even more as a flight simulator for music landscapes.

Another future application scenario concerns mobile devices. We are developing a version of the nepTune interface that Java 2 Mobile Edition enabled devices can execute. While a personal computer must perform the audio feature extraction step, it's possible to perform the remaining steps—that is, SOM training and landscape creation—on the mobile device. Considering the ongoing trend toward mobile music applications and the necessity of simple interfaces to music collections, the nepTune interface could be a useful and fun-to-use approach for accessing music on portable devices. Figure 5 shows a screen shot of the current prototype.

We can conceive of many alternative ways of accessing music on mobile music players. For instance, we have recently developed another interface, also based on automatic music similarity analysis, that permits the user to quickly locate a particular music style by simply turning

a wheel, much like searching for radio stations on a radio.⁹ Figure 6 shows our current prototype, implemented on an Apple iPod, in which the click wheel helps navigate linearly through the entire music collection, which the computer has arranged according to musical similarity. A number of other research laboratories are also working on novel interfaces.¹⁰⁻¹²

In general, we believe that intelligent music applications (of which nepTune just gives a tiny glimpse) will change the way people deal with music in the next few years. Computers that learn to understand music in some sense will become intelligent, reactive musical companions. They will help users discover new music; provide informative meta-information about musical pieces, artists, styles, and relations between these; and generally connect music to other modes of information and entertainment (text, images, video, games, and so on). Given the sheer size of the commercial music market, music will be a driving force in this kind of multimedia research. The strong trend toward Web 2.0 that floods the Web with texts, images, videos, and audio files poses enormous technical challenges to multimedia, but also offers exciting new perspectives. There is no doubt that intelligent music processing will become one of the central functions in many future multimedia systems. **MM**

Acknowledgments

The Austrian Science Fund under FWF project number L112-N04 and the Vienna Science and Technology Fund under WWTF project number CI010 (Interfaces to Music) support this research. We thank the students who implemented vital parts of the project, especially Richard Vogl, who designed the first interface prototype; Klaus Seyerlehner, who implemented high-level feature extractors; Manfred Waldl, who created the prototype of the mobile nepTune version; and Dominik Schnitzer, who realized the intelligent iPod interface.

References

1. E. Pampalk, "Islands of Music: Analysis, Organization, and Visualization of Music Archives," master's thesis, Vienna Univ. of Technology, 2001.
2. M. Mandel and D. Ellis, "Song-Level Features and Support Vector Machines for Music Classification," *Proc. 6th Int'l Conf. Music Information Retrieval (ISMIR)*, 2005, pp. 594-599; <http://ismir2005.ismir.net/proceedings/1106.pdf>.



Figure 6. Prototype of our intelligent music wheel interface, implemented on an iPod.

3. J.J. Aucouturier, F. Pachet, and M. Sandler, "The Way It Sounds: Timbre Models for Analysis and Retrieval of Music Signals," *IEEE Trans. Multimedia*, vol. 7, no. 6, 2005, pp. 1028-1035.
4. T. Kohonen, *Self-Organizing Maps*, vol. 30, 3rd ed., Springer Series in Information Sciences, Springer, 2001.
5. W.P. Tai, "A Batch Training Network for Self-Organization," *Proc. 5th Int'l Conf. Artificial Neural Networks (ICANN)*, vol. II, F. Fogelman-Soulié and P. Gallinardi, eds., EC2, 1995, pp. 33-37.
6. E. Pampalk, A. Rauber, and D. Merkl, "Using Smoothed Data Histograms for Cluster Visualization in Self-Organizing Maps," *Proc. Int'l Conf. Artificial Neural Networks (ICANN)*, Springer LNCS, 2002, pp. 871-876.
7. P. Knees et al., "Automatically Describing Music on a Map," *Proc. 1st Workshop Learning the Semantics of Audio Signals (LSAS)*, 2006, pp. 33-42; http://irgroup.cs.uni-magdeburg.de/lsas2006/proceedings/LSAS06_Full.pdf
8. K. Lagus and S. Kaski, "Keyword Selection Method for Characterizing Text Document Maps," *Proc. 9th Int'l Conf. Artificial Neural Networks*, vol. 1, IEEE Press, 1999, pp. 371-376.
9. T. Pohle et al., "'Reinventing the Wheel': A Novel Approach to Music Player Interfaces," *IEEE Trans. Multimedia*, vol. 9, no. 3, 2007, pp. 567-575.
10. M. Goto and T. Goto, "Musicream: New Music Playback Interface for Streaming, Sticking, and Recalling Musical Pieces," *Proc. 6th Int'l Conf. Music*

Information Retrieval (ISMIR), 2005, pp. 404-411; <http://ismir2005.ismir.net/proceedings/1058.pdf>.

11. E. Pampalk and M. Goto, "MusicRainbow: A New User Interface to Discover Artists Using Audio-Based Similarity and Web-Based Labeling," *Proc. 7th Int'l Conf. Music Information Retrieval* (ISMIR), 2006, pp. 367-370; http://ismir2006.ismir.net/PAPERS/ISMIR0668_Paper.pdf
12. R. van Gulik, F. Vignoli, and H. van de Wetering, "Mapping Music in the Palm of your Hand: Explore and Discover your Collection," *Proc. 5th Int'l Conf. Music Information Retrieval* (ISMIR), 2004, pp. 409-414; <http://ismir2004.ismir.net/proceedings/p074-page-409-paper153.pdf>



Peter Knees is a project assistant in the Department of Computational Perception, Johannes Kepler University Linz, Austria. His research interests include music information retrieval, Web mining, and information retrieval.

Knees has a Dipl. Ing. (MS) in computer science from the Vienna University of Technology and is currently working on a PhD in music information retrieval from Johannes Kepler University Linz.



Markus Schedl is working on his doctoral thesis in computer science at the Department of Computational Perception, Johannes Kepler University Linz. His research interests include Web mining, (music) information retrieval, information visualization, and intelligent user interfaces. Schedl graduated in computer science from the Vienna University of Technology.



Tim Pohle is pursuing a PhD at Johannes Kepler University Linz, where he also works as a research assistant in music information retrieval with a special emphasis on audio-based techniques. His research interests include musicology and computer science. Pohle has a Dipl. Inf. degree from the Technical University Kaiserslautern, Germany.



Gerhard Widmer is a professor and head of the Department of Computational Perception at the Johannes Kepler University Linz, and head of the Intelligent Music Processing and Machine Learning Group at the Austrian Research Institute for Artificial Intelligence, Vienna. His research interests include machine learning, pattern recognition, and intelligent music processing. Widmer has MS degrees from the University of Technology, Vienna, and the University of Wisconsin, Madison, and a PhD in computer science from the University of Technology, Vienna. In 1998, he was awarded one of Austria's highest research prizes, the Start Prize, for his work on AI and music.

Readers may contact Peter Knees at the Dept. of Computational Perception, Johannes Kepler University Linz, Altenberger Str. 69, 4040 Linz, Austria; peter.knees@jku.at.

For further information on this or any other computing topic, please visit our Digital Library at <http://computer.org/publications/dlib>.

Now
available!

FREE Visionary Web Videos
about the Future of Multimedia.

Listen to premiere multimedia experts!
Post your own views and demos!

Visit www.computer.org/multimedia