ADAPTIVE MULTIMODAL EXPLORATION OF MUSIC COLLECTIONS

Dominik Lübbers[*†], **Matthias Jarke**[*]
[*]Informatik 5
RWTH Aachen University
Aachen, Germany
[†]Dept. Applied Information Technology
German University of Technology
Muscat, Sultanate of Oman

ABSTRACT

Discovering music that we like rarely happens as a result of a directed search. Except for the case where we have exact meta data at hand it is hard to articulate what song is attractive to us. Therefore it is essential to develop and evaluate systems that support guided exploratory browsing of the music space.

While a number of algorithms for organizing music collections according to a given similarity measure have been applied successfully, the generated structure is usually only presented visually and listening requires cumbersome skipping through the individual pieces.

To close this media gap we describe an immersive multimodal exploration environment which extends the presentation of a song collection in a video-game-like virtual 3-D landscape by carefully adjusted spatialized plackback of songs. The user can freely navigate through the virtual world guided by the acoustic clues surrounding him.

Observing his interaction with the environment the system furthermore learns the user's way of structuring his collection by adapting a weighted combination of a wide range of integrated content-based, meta-data-based and collaborative similarity measures.

Our evaluation proves the importance of auditory feedback for music exploration and shows that our system is capable of adjusting to different notions of similarity.

1. INTRODUCTION

Early work in Music Information Retrieval primarily concentrated on the development and evaluation of systems to support the identification of songs in a collection given a well-formulated query. According to Cunningham [1], this retrieval paradigm hardly matches the way we usually look for CDs in a music shop. Instead of searching for a dedicated album, participants in a user study showed a more exploratory browsing behaviour, which can be summarized as "shopping around" in contrast to "shopping for". This exploratory behaviour is however not completely chaotic: Users are reported to prefer some sort of structure in a music collection (e.g. a categorization according to genres), as long as this organization is intuitively understandable to them.

Even having a specific song in mind, we may find it difficult to articulate the information demand properly, if the name of the artist and the song title are unknown. Query by Example approaches like Query by Humming can only partly bridge this media discontinuity gap.

These reasons have led to an increased interest in exploration environments for music over the last years [2–4]. Most of theses approaches focus on visualizing a music collection with only standard playback functionality, which results in a media discontinuity problem in the opposite direction and does not exploit the human's capability to orientate himself in a complex environment of simultaneously playing spatialized sounds.

Therefore, we developed and evaluated an exploration prototype that provides an immersive virtual environment, in which the user can navigate guided by acoustic clues from song playbacks surrounding him.

As in previous approaches, the placement of pieces in this environment is based on a similarity function. The notion of similarity is known to be multifaceted, highly user-dependent and also influenced by the song collection at hand. We therefore allow the user to move songs in the environment as well as raise or lower borders between song clusters. Observing the user's interaction with the landscape we furthermore adapt a linear combination of content-based and collaborative similarity measures to best fit his understanding of similarity.

To our knowledge our prototype is therewith the first multimodal exploration environment which integrates an immersive virtual 3D-landscape of clustered songs with spatialized audio playback respecting humans' auditory perception limitations and furthermore adapts to the user's strategy of organizing his collection by learning the weights of a wide range of different music similarity measures.

In the next section we give a brief overview of related work on exploration environments for music collections. Then we list the integrated base similarity functions used as components of a user-adaptive similarity measure. The following section describes our exploration environment in detail. We continue with the explanation of the similarity measure adaption process, which is followed by results from an qualitative and quantitative evaluation of our system and concluded by some final remarks and an outlook on further research.

2. RELATED WORK

Over the last years, a number of proposals for visualizing music collections have been made.

Pampalk et al. reduce the audio signal of a song to the median of frame-based *Fluctuation Patterns*, which model loudness periodicities in different frequency bands of the signal [5]. These features are used to train a small-size rectangular Self-Organizing Map (SOM). They interpret the estimated song densities of the cells as the height profile of a map. Applying an appropriate color map generates an intuitive visualization of similar song clusters positioned on "Islands of Music" separated by blue water.

The approach by Moerchen et al. is conceptually similar [7]. Their work mainly differs in the use of a compact but highly discriminative content-based feature set and the distribution of the collection items over a larger, *emergent* SOM. Still, Moerchen et al. do not integrate any kind of acoustic presentation besides a standard playback functionality of a selected song.

In contrast to this, Hamanaka and Lee focus on audio-only exploration of a given song set [8] without the need for a display. By spatializing songs according to different pre-defined allocation schemes, a user wearing a special headphone has the impression of being surrounded by simultaneously playing sound sources from different directions. Sensors mounted on the headphone detect the movement of the head and allow the user to change focus to songs he perceives from left or right. This interaction promotes the impression of an immersive virtual environment. Additionally, he can narrow the range of sounding sources by putting his hands behind the ear and thereby fading out songs that are not placed directly in front of him. This resembles the *focus of perception* mechanism we introduced in [9] and supports humans' ability to concentrate on specific sounds in a complex mixture, known as the cocktail party effect.

To our knowledge, the approach by Knees et al. is the first one that combines SOM-based structuring of music collections with three-dimensional visualization and auralization to an immersive multimodal exploration environment [10]. Their work extends the Island of Music metaphor by using the smoothed height profile of SOM cells to generate a virtual 3D-landscape that the user can intuitively explore. Songs in the neighborhood of the current position sound from the respective direction. Knees and et. do not implement a focus mechanism, which seems to be critized by one of the comments in their user study, that asks for a larger landscape especially when facing crowded regions.

All of the above exploration environments quantify similarity between songs according to a fixed measure, that is supposed to reflect a generic similarity understanding by the average user. Recognizing the diversity of the similarity notion, Pampalk et al. align three SOMs representing timbral, rhythmic and metadata-provided aspects and allow the user to gradually change between these presentations [11].

Baumann linearly combines content-based similarity with cultural similarity and text-based similarity of the lyrics [12]. The user can adjust the weights of this trimodal measure by moving a virtual joystick into the direction of the favoured similarity aspect.

Instead of forcing the user to learn the semantics of different similarity measures and to decide for the individual importance of them, we propose a machine learning strategy that induces the weights of each component from the user's interaction with our immersive multimodal exploration environment.

Figure 1 depicts the stages involved in generating and adapting this environment. The following sections describe these phases in detail.

3. SIMILARITY

To model a user's notion of similarity as precisely as possible, it is mandatory to combine a number of base similarity measures covering different musical aspects and let the system adapt their weights.

We therefore decided to integrate timbral similarity measures (based on stochastic MFCC models as proposed by Logan/Salomon [13] and Aucouturier/Pachet [14] or the 20-feature set proposed by Moerchen et al. [7]) as well as more rhythm-based measures (Fluctuation Patters and Periodicity Histograms [11]). Furthermore we calculate the average and variance of 15 frame-based audio features as provided by the MIRtoolbox library [15]. These features are of varying complexity, ranging from simple RMS values over spectral centroids and roughness measures to key clarity and tempo estimates.

Additionally, we use ID3 metadata to make contextual information available. In particular, we calculate the time period between the publication of two pieces. To group songs by the same artist even in the commonly encountered presence of small typing errors, we furthermore calculate the edit distance between ID3 artist strings.

These similarity measures are complemented by three collaborative approaches based on direct last.fm similarity links, last.fm top tags and co-occurrence on playlists published on Art of the Mix.

last.fm offers the compilation of recommended tracks to a personalized music stream based on the user's profile. This requires the establishment of similarity links between tracks. last.fm allows access to this information by a web service that returns a number of similar tracks to a given song. Each of these similar tracks is assigned a match value that quantifies the degree of similarity scaled to 100 for the most similar song. We consider the presence of a direct similarity link as a strong indication of similarity, even if the match value might be low. Therefore we transform the match score with a compressed exponential function to a distance value. Averaging the mutual distances to guarantee symmetry leads to the following calculation for two tracks tr_i and tr_j:

$$d_{DL}(tr_i, tr_j) = 0.5(e^{-c_{DL} \cdot \frac{\mathrm{ms}_{tr_i}(tr_j)}{100}} + e^{-c_{DL} \cdot \frac{\mathrm{ms}_{tr_j}(tr_i)}{100}}),$$

where $\mathrm{ms}_{tr_i}(tr_j)$ denotes the match score of track tr_j in

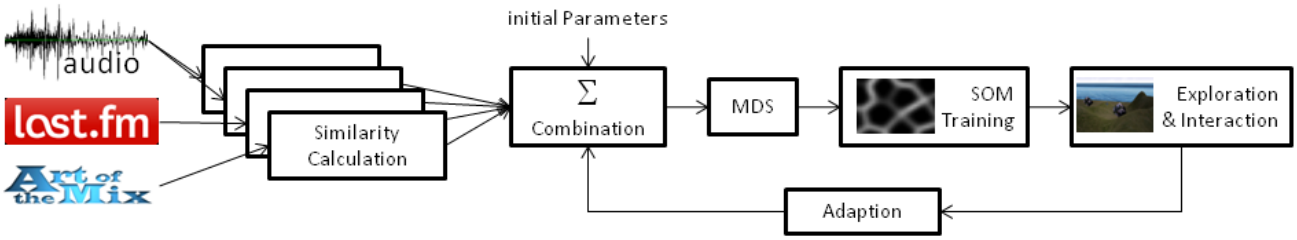

Figure 1. Data transformation stages for building and adapting the exploration environment.

the list of similar tracks to track tr_i if present and 0 otherwise. We empirically chose a value of $c_{DL} = 5$ for the compression factor.

While a track-based similarity measure is very specific, it may be difficult to find enough collaborative data for a reliable estimate. We therefore calculate the distance between the artists of two songs in the same way as above and combine it linearly with the track-based measure weighting the more precise track distance double.

Instead of assigning fixed genre categories to songs, last.fm allows users to tag tracks with arbitrary keywords favouring the emergence of a folksonomy over the definition of a static genre hierarchy. The comparison of these song descriptions is another valueable source of similarity. Retrieving the top tags for a song results in a list ranked according to the frequency used to annotate the song. Unfortunately, last.fm's `count` attribute does not quantify this per-track frequency but the overall popularity of a tag. Lacking further information, we consider the tags as natural language terms in a text about the track. This allows us to assume that the tag distribution follows Zipf's law and approximate tag frequencies by a Zipfian density function. Likewise, we do not have access to the ratio of tracks that are tagged with a certain keyword and have to estimate the inverse document frequency on the basis of the overall popularity of a tag.

These approximations can be used to weight the importance of a tag for a song according to the standard tf·idf scheme. The track-based top tag-similarity between two songs can finally be calculated as the cosine between aligned weight vectors. For the same reasons as above we also calculate top tag-similarity on artist level.

The last distance calculation we derive from collborative data is based on co-occurrences of songs on playlists (called *mixes*) that are published by users on the Art of the Mix portal [1]. We follow the assumption that two pieces occuring on the same list fit the same taste and can be considered as similar. To quantify this notion we use a simple overlap distance measure:

$$d_{AotM}(s_i, s_j) = 1 - \frac{|M(s_i) \cap M(s_j)|}{\min\{|M(s_i)|, |M(s_j)|\}},$$

where $M(s_i)$ denotes the set of mixes that contain song s_i. As done for the other collaborative measures, we combine this distance with its artist-based variant.

[1] www.artofthemix.org

Since some of the presented measures (like Logan/Salomon) are based on pairwise comparisons between songs, the composed distance values are arranged in a (symmetric) matrix. As the SOM training algorithm requires the representation of each item as a feature vector in Euclidean space, we apply multi-dimensional scaling (MDS) to find d-dimensional coordinates for each song such that the Euclidean distance between two song vectors resembles the distance matrix value (see figure 1). In our experiments we chose a value of $d = 20$, which matches the dimensionality of the data space used for the MusicMiner-SOM [7].

4. EXPLORATION ENVIRONMENT

4.1 SOM Training

As humans are used to intuitively estimate distances between points on a 2-dimensional plane, dimensionality reduction techniques that map high-dimensional data to low-dimensional representations while preserving distances as much as possible are popular data visualization strategies.

One of these techniques is the Self-Organizing Map (SOM) proposed by Kohonen,which arranges disjoint cells $\{y_i\}$ on a usually rectangular grid. Each y_i is associated with a *model vector* m_i from data space. We initialize the model vectors with linear combinations of the first two principial components of the song feature values according to the grid coordinate of their cell.

In each iteration t we randomly choose a data vector x_j and identify the cell bm with the closest model vector to x_j, i.e. that minimizes $||x_j - m_{bm}||$. The model vectors of this *Best Matching Unit* bm and its neighborhood are moved towards x_j according to the following equation:

$$m_i(t+1) = m_i(t) + \alpha(t) \cdot h_{i,bm}(t)[x_j - m_i(t)],$$

where $\alpha(t)$ denotes the learning rate at time t and $h_{i,bm}(t)$ quantifies the distance between x_i and bm, usually by some Gaussian function centered around bm. Since $\alpha(t)$ and $h_{i,bm}(t)$ decrease with each iteration and thereby weaken the adaption process with time, the map converges to a configuration where the Best Matching Units of similar data points are located close to each other.

In contrast to clustering algorithms like k-Means, a SOM is also capable of adaquately representing data points that lie in between clusters and reveals the macro-structure of the data space by retaining similarity relationships between clusters themselves.

The distribution of model vectors over the grid that is generated on the fly during the adaption contains additional valueable information about the similarity space: This information can be visualized by the U-Matrix [6], which assigns to each cell the average distance of its model vector to the model vectors of its neighbors. High values thereby indicate clear borders separating coherent regions of similar objects on the map.

4.2 Visual Presentation

Displaying these U-Matrix values and placing songs at their Best Matching Unit already yields an untuitively understandable visualization of the collection. But if we interpret the U-Matrix values as heights of a landscape we can generate a 3-D terrain and allow the user to leave his bird's eye-view on the music space in favor of becoming part in an immersive virtual environment.

Our prototype is based on Microsoft's game framework XNA 3.0 to realize efficient state-of-the-art visualization. We generate a high-resolution terrain mesh by bilinear interpolation of the U-Matrix height values and use a customized shader for visualization which appropriately combines sand, grass, mountain and snow textures according to the height.

By default, songs are visualized as small cubes textured with the cover image of their album if available. The position of a cube is mainly determined by the coordinates of the song's Best Matching Unit. To avoid clumping at grid points, we slightly move it towards the location in the immediate neighborhood where the bilinearly interpolated model vector is closest to the feature vector of the song.

The user can freely run through the landscape, move his head around and lift up to get an overview of the scenery. Figure 2 shows a screenshot of our environment taken from different elevation levels. The user is standing in (or over) a valley that contains songs from the German hiphop group *Fanta4*. As can be seen, these songs are clearly separated from different pieces by surrounding hills.

4.3 Auditory Presentation

Music is described best by music. This asks for the presence of acoustic information as guidance in the exploration process: Since humans are used to differentiate well between sound sources from different directions, exposing the user to simultaneously playing spatialized music facilitates efficient and well-informed navigation through the collection.

Fortunately, the above virtual environment can be extended naturally to incorporate the presentation of acoustic information, simply by associating each cube with a sound source playing the song from its location in the landscape.

As described in [9] the unrestricted simultaneous playback of many songs quickly overwhelms the user's auditory system and confuses more than it helps. Following ideas from visual perception we therefore define the point the user is currently looking at as the *Focus of Perception* and attenuate the volume of songs the more they deviate from the view direction. To allow for broad "listening around" as well as for clearly focussing on the sound in front we model the strength of this attenuation by a Gaussian function with user-adjustable variance. More precisely, the gain factor due to perception focussing is given as follows:

$$g_{PF}(\varphi) = e^{\frac{-\varphi^2}{\sigma^2}},$$

where φ denotes the angle between the direction to the song and the view direction and $\sigma^2 = \frac{-AoP}{\ln(gAoP)}$ is the variance for the user-adjustable *Angle of Perception* AoP, such that $g_{PF}(AoP) = gAoP$.

We describe the influence of a song's distance to its gain by an inverse distance model:

$$g_{Dist}(d) = \min(1, \frac{decSpeed}{d} - \frac{decSpeed}{minDist} + 1),$$

where d is the distance to the song, $decSpeed$ parameterizes the speed of gain decrease per distance unit and $minDist$ denotes the distance at which no attenuation takes place.

To summarize, the overall gain for a song s at location $\vec{p_s}$ assuming a listener's position $\vec{p}$ and a view direction $\vec{vd}$ is the product of its gain influences:

$$g(s,\vec{p}) = g_{Dist}(||\vec{p}-\vec{p_s}||)\cdot g_{PF}(\angle(\vec{vd}, \vec{p_s}-\vec{p}))\cdot g_{muff}(s,\vec{p}).$$

$g_{muff}(s,\vec{p})$ reduces the gain for a song, that is hidden behind a rise of the terrain. To generate the impression of a muffled sound this is complemented by a highcut filter.

Still, the simultaneous playback of all songs in the collection is too demanding (as well from an computational as from a perceptual point of view). We tested several song selection criteria and decided for a simple approach that guarantees perceptual separability and does not change the set of active sources when the user rotates his head: First, all songs in the neighborhood of the listener's position are sorted according to their gain factor. Following this order we then successively activate songs as long as they do not sound from a direction similar to the one of already playing songs.

4.4 User Interaction

A standard xBox 360 game controller can be used to navigate in the virtual world. Besides this, the user can customize the landscape as follows:

- Songs that seem to be misplaced in the opinion of the user can be moved easily.
- Alternatively, songs can be released to let the system find a new location during the next adaption cycle.
- Landmarks can be placed to emphasize and easily recover locations on the terrain. The user can choose between different sign types that can be labeled or textured with arbritary images. Figure 2 shows two triangular landmarks.
- The terrain can be altered by raising or lowering its height at the position the user points to. This allows the formation of new separating hills between song

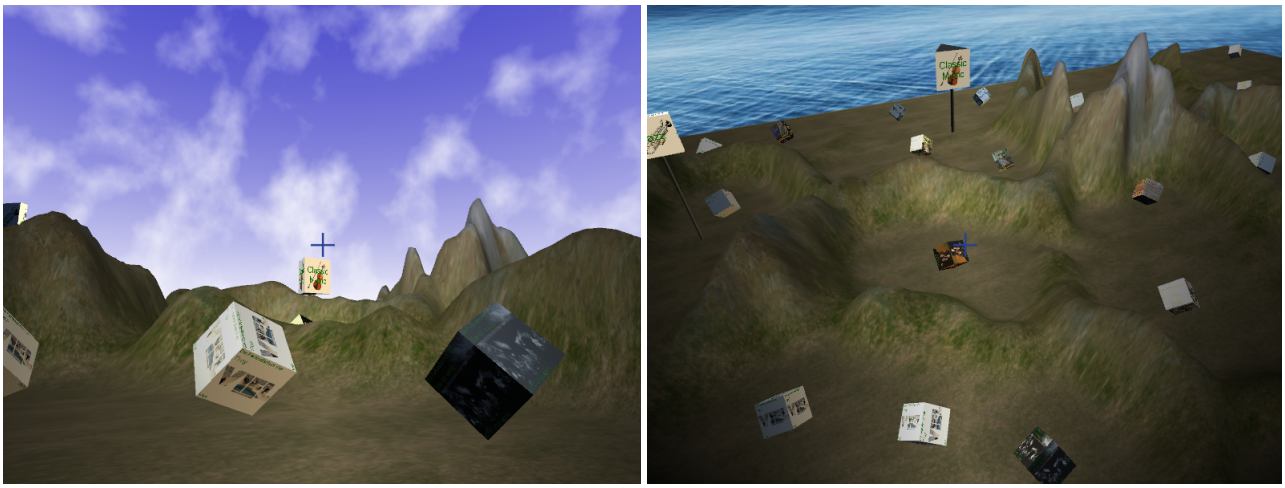

Figure 2. Screenshot of the exploration prototype: Views from different elevation levels

clusters that are perceived as different or the removement of borders between areas that the user judges similar.

5. USER ADAPTATION

As Cunningham observes, music listeners organize their personal collections according to different criteria. Some may sort their albums by the year of publication, some may cluster their music by genre, for others rhythmic content plays a dominant role. An exploration environment should be flexible enough to follow the user's organization strategy.

Instead of asking the user to articulate his structure principles explicitly we decided to learn his similarity notion from his interaction with the environment. Adapting the weights in the linear similarity model properly allows us to reposition songs that have been released by the user or to place new songs that are added to the collection.

The user can build or destroy separating hills between songs. To account for these terrain changes, we numerically integrate over the height profile (h_n) between the locations p_i and p_j and compare this to the situation before the change (h_o):

$$td_t(s_i, s_j) = \frac{1}{||\vec{p_i} - \vec{p_j}||} \cdot \left(\int_{\vec{p_i}}^{\vec{p_j}} (h_n(\vec{p}) - h_o(\vec{p}))d\vec{p} \right)$$

The combination of td_t with the Euclidean distance between the (interpolated) model vectors of two songs' locations on the map is stored in a *target distance* matrix. Each entry of this matrix is considered a training case for a linear regression learner, that adapts the weighting of the implemented base distances to approximate the target distance.

As figure 1 shows, the updated similarity model is subsequently used to rebuild the environment by the same process chain as before. To avoid drastic changes in the exploration space that potentially disorientate the user, we initialize the vector representation of each fixed song by its old value before the MDS optimization starts. Likewise, we guarantee topographic stability of the SOM by constantly taking a song's old location as its Best Matching Unit during training.

6. EVALUATION

We conducted a user study with nine participants showing different music taste, listening habits and experience with computer games.

In a first experiment we aurally presented an unknown song and measured the time needed to find it in a collection of about 100 tracks, that were randomly distributed over a flat exploration plane. Cover and metadata of the wanted song were not given to the user. We repeated the task for a different song and collection, this time providing the SOM-based organization. To eliminate effects from the choice of song and collection, we shuffled task and data for different participants.

A similar pair of experiments investigates the importance of spatialized acoustic clues when navigating through the exploration space by comparing this feature with standard media player functionality which requires to explicitly start and stop the playback of a song.

We found reductions in search time of 61% and 58% on average, which demonstrate, how significantly the user benefits from a well structured collection and acoustic clues during the exploration.

The last group of experiments evaluate the adaptation capabilities of our system to a user's notion of similarity: We asked the participants to customize a collection of 20 tracks by moving the songs and changing the terrain structure. Similar to a leave-one-out evaluation we successively release one song and compare its original position to the location that would be assigned by the SOM training. This *placement error* is calculated with and without executing the adaptation procedure. The first data series in figure 3 shows the relative difference between these two runs and reveals, that generally the adaptation works well, but reduces the placement error only slightly. One reason for that might be the that the initial similarity measure already captured the user's notion rather well.

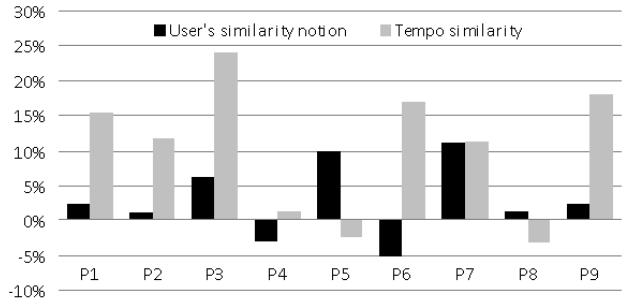

Figure 3. Relative reduction of placement error by adaptation to users' similarity notion

Therefore, we asked the users to organize the collection according to tempo independent of the genre and again computed the relative improvement in placement error. As can be seen from the second data series in figure 3 our system also adapts generally well to this more drastic change in similarity notion.

After these quantitative experiments we handed out an extensive questionnaire for qualitative evaluation. Study participants consistently judged the usability of the system as high but repeatedly proposed the addition of a 2-D map view to the environment to avoid disorientation in the exploration landscape.

7. CONCLUSION AND OUTLOOK

We presented an immersive multimodal exploration environment, that visualizes and auralizes music collections organized according to an user-adaptable similarity model, which combines content-based, meta-data-based and collaborative similarity measures. While our evaluation shows the general tractability of our approach, some open questions for further research remain:

So far, we did not focus on scalability issues in our work. We found, that collections of up to 400 songs are still manageable in our environment. Larger numbers of tracks require some form of hierarchical organization to remain accessible. We may can adopt ideas from [16] to extend the SOM-based placement algorithm.

Since they can model more complex relationships than vector-based distances, we deliberately integrated similarity measures that require pairwise computation of distances. Because of this the complexity of the similarity calculation stage is in $\mathcal{O}(n^2)$. To alleviate the scalability problems arising from this, one could restrict the calculation to some *anchor songs*. The MDS stage is already prepared to handle sparse distance matrices.

As shown by the evaluation, the adaption to the user's similarity notion still has room for improvement. A reason for this might be that a linear model is not expressive enough to capture the intended combination of base similarities. More complex models should therefore be investigated in future research.

8. REFERENCES

[1] S. Cunningham, N. Reeves, and M. Britland: "An ethnographic study of music information seeking: implications for the design of a music digital library" *JCDL '03: Proceedings of the 3rd ACM/IEEE-CS Joint Conference on Digital Libraries*, pp. 5–16, 2003.

[2] M. Goto and T. Goto: "Musicream: New Music Playback Interface for Streaming, Sticking, Sorting, and Recalling Musical Pieces" *Proc. ISMIR*, 2005.

[3] R. van Gulik, F. Vignoli, and H. van de Wetering: "Mapping Music in the Palm of Your Hand, Explore and Discover Your Collection" *Proc. ISMIR*, 2004.

[4] E. Pampalk and M. Goto: "Musicsun: A New Approach to Artist Recommendation" *Proc. ISMIR*, 2007.

[5] E. Pampalk, A. Rauber, and D. Merkl: "Content-based Organization and Visualization of Music Archives" *Proceedings ACM Multimedia*, 2002.

[6] A. Ultsch: "Self-Organizing Neural Networks for Visualization and Classification" *Proc. GfKI*, 1992

[7] F. Mörchen, A. Ultsch, M. Nöcker, and C. Stamm: "Databionic Visualization of Music Collections According to Perceptual Distance" *Proc. ISMIR*, 2005.

[8] M. Hamanaka and S. Lee: "Music Scope Headphones: Natural User Interface for Selection of Music" *Proc. ISMIR*, 2006.

[9] D. Lübbers: "soniXplorer: Combining Visualization and Auralization for Content-Based Exploration of Music Collections" *Proc. ISMIR*, 2005.

[10] P. Knees, M. Schedl, T. Pohle, and G. Widmer: "Exploring Music Collections in Virtual Landscapes" *IEEE MultiMedia*, Vol. 14, No. 3, 2007.

[11] E. Pampalk, S. Dixon, and G. Widmer: "Exploring Music Collections by Browsing Different Views" *Proc. ISMIR*, 2003.

[12] S. Baumann, T. Pohle, and S. Vembu: "Towards a Socio-cultural Compatibility of MIR Systems" *Proc. ISMIR*, 2005.

[13] B. Logan and A. Salomon: "A Music Similarity Function Based on Signal Analysis" *Proceedings ICME*, 2001.

[14] J.-J. Aucouturier and F. Pachet: "Finding Songs That Sound the Same" *IEEE Workshop on Model based Processing and Coding of Audio* 2002.

[15] O. Lartillot and P. Toiviainen: "A Matlab Toolbox for Musical Feature Extraction From Audio" *Proc. DAFx-07*, 2007.

[16] A. Rauber, E. Pampalk, and D. Merkl: "Using psycho-Acousic Models and Self-Organizing Maps to Create a Hierarchical Structuring of Music by Sound Similarity" *Proc. ISMIR 2002*