Music Information Retrieval & Visualization

Tim Langer

Abstract—The growing possibilities for digital storage has led to large personal music collections that require new technologies to use and maintain them. There has been a lot of research done related to visualizing music collections in order to improve browsing, exploring and searching of music. While almost every publication in this subject has its own purpose and approach to achieve it there still exist common problems, ideas and methods. In the following I will identify major causes that led to such development, based on state of the art technology, at first. I will then further specify a couple of commonly used ideas to solve those problems as well as techniques to realise them. At last a couple of chosen examples are presented to demonstrate approaches to the prior specification. The ambition of this paper is to identify the development within Music Information Retrieval & Visualization and present a survey of recent research.

Index Terms—MIR, Music, Information Visualization, Music Collections, Music Visualization, Music Similarity

1 Introduction

With the growing impact of technology on everyday life the research field of Information Visualization has developed the new and important topic of Personal or Casual Information Visualization [24] which puts the focus on technologies that try to include visualization into the common daily life cycle. The focus of this paper will be on the domain of music collections by clarifying the current problems within wide-spread music players and giving an overview about recent research that tries to solve those problems and add new functionality to support the user. The next section will summarize existing problems while the third and fourth section will feature new concepts to adopt to the growing demand for development in terms of strategy, ideas and visualization. The fifth section lists a couple of exemplary research and at the end conclusions will be drawn.

2 Status Quo

In today's world digital music collections are usually organized based on a selfmade system by the owner of the collection. Such organisation systems usually vary a lot in their structure - one might sort his collection by artist, by album or by the date of release. Influenced by those heterogeneous structures the current methods for displaying and organzing music in state of the art digital music players [19] are playlists created by the user. People tend to fill such playlists with similar music to create playlists for specific emotional states (for example slow music that helps to relax). This human necessity is one of the main reason why new researches in MIR topic usually rely on similarity-measures to arrange music collections and it has been proved to be a well working concept ([19] [31] [26]).

2.1 Problems

As stated above the tools to listen to music are typically commercial products with a large spread. With the growing size of a music collection it gets harder to find the music you are looking for or simply browse your collection. As the only possibility to search the music, through the file system of the operating system, is based on text, the user has to know at least some part of the file's title to actually find it through the search. When thinking about use cases as described in section 3.1 this gives no opportunity at all to get results related to music a user already knows which is a basic demand. To solve this problem meta data is appended to the digital music to provide further information. The thereby developed problems will be explained by using genres as an example in section 2.1.2

- *Tim Langer is studying Media Informatics at the University of Munich, Germany, E-mail: tim.langer@campus.lmu.de*

- *This research paper was written for the Media Informatics Advanced Seminar on Information Visualization, 2008/2009*

2.1.1 Playlists

As stated above the basic systems used nowadays rely on playlists to visualize music collections. But as pictures are easier and faster to recognise for a human, it is quite intuitive that such would be a better choice than text based methods. [31] states this as following: "However this [text] approach by itself is inadequate for effective search and retrieval". Also it is quite hard to put a lot of information into a short text whilst a graphical visualization could present different information by varying itself in terms of size, colour, form or other parameters. On going with a growing music collection and a lot of cross-links information (such as different artists on the same song or one song on different albums) a large amount of playlists is needed to keep up with all this and thereby the clear view gets lost. As mentioned before, some of those music players already try to create automatic playlists. This is done by either studying the users listening behaviour, and grouping favourite tracks together, or by analyzing the metadata. The thereby extracted information is then used to create playlists for different categories, usually distinguished by the genre (see 2.1.2). L As listening to music is correlated to emotions the choice of music tends to depend on our current mood [27]. So searching music that fits this mood would be a quite intuitive feature! But with playlists this is only possible if the user knows what kind of music is stored in a playlists and/or if he already created a playlists that fits this mood. So the system hardly aids the user with his choice.

2.1.2 Tags

The adding of meta information with ID3-Tags brings a whole lot of new problems with it. Firstly, tags provide information appended to and not derived from the music which therefore can obviously contain false information. Secondly, as such tags are added by communities of voluntary users stored in large online databases faults (such as typing errors) are inevitable. Thirdly, the process of assigning the metadata to a musical track is a subjective process ([31] [19]). One person might classify a song simply as "Rock" while another person might go into more detail and categorise it as "Hard-Rock". Hilliges et. al.[19] provide a good example for this when stating that the well-known iTunes music store puts both punk-rockers Anti-Flag and musician James Blunt into the category of Rock. But not only the subjective assignment of Genres (and other tags) is a problem, also the number of possibilities that can be used is problematic field. On the one hand when specifying too many different choices (such as classifying dozens of different subcategories for Rock) the large amount of details makes it almost impossible to still maintain an informational overview. On the other hand tagging is almost futile when putting all your data into the same categories as it provides no further information. And last but not least most musicians develop different styles throughout their musical career. It might usually be possible to categorize one album into a matching genre, but rarely the whole artist based on all his or her releases. And sometimes it even impossible to sum up the different songs in one album into one matching genre as the style of the tracks might vary because of different influences, featured artists or for other reasons. In fact "few artists are truly a single 'point' in any imaginable stylistic space but undergo changes throughout their careers and may consciously span multiple styles within a single album, or even a single song" [2]

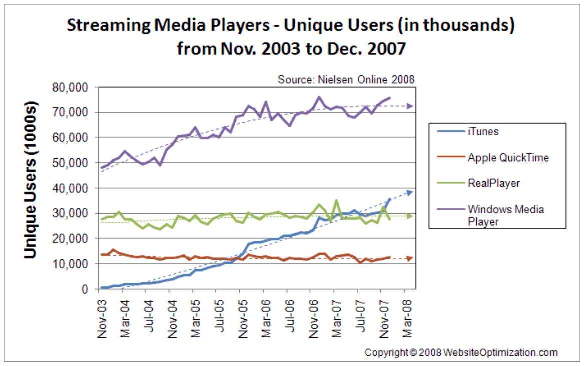

Fig. 1. Streaming Media Players - Unique Users by www.websiteoptimization.com

2.2 Examples

The following section will introduce three state of the art music and media players. They were chosen as a representative of the three biggest operating systems: Microsoft's Windows, Apple's Mac and the Linux system. The Windows Media Player and iTunes also posses a large market share within digital media players.

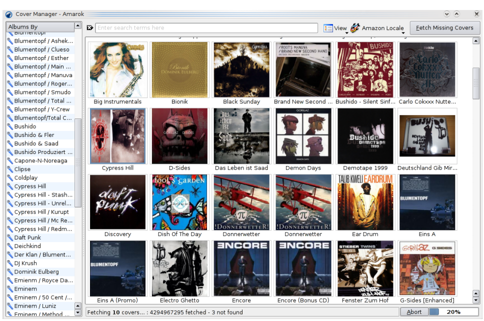

Fig. 2. Visualization using Album Covers (example from Amarok)

2.2.1 Windows Media Player (WMP)

Due to the enormous market share of the Miccosoft operating system Windows [1], the Windows Media Player enjoys a market share of about 50% [18] (and more than 70 million unique streaming users *(see figure 1))*. Currently in its 11th version, it sticks to established structures like playlists and only takes partial advantage of the findings by new research in this topic. But also some of them were taken into account and so WMP offers integration of several (user-made) plugins (for example new track-wise visualizations). The visualization of whole albums *(see figure 2)* is possible and done by presenting the album cover (taken from online databases or added manually by the user). As this also relates to the physical apperance of a music album it is not the worst choice, but still has weaknesses. As said, if the album cover is not available there is nothing but a default image to display unless the user adds the cover manually. The user has the choice between several types of track-wise visualizations *(see figure 4)* that can be added as a plugin and even manually created using a SDK. Grouping of similar music is realised with a stack metaphor *(see figure 5) (top)*.

[1]80%-90% measured by `http://reseller.co.nz/` at 05.01.2009

2.2.2 Amarok

Amarok[2] is a Linux based music player. Just like the Windows Media Player it uses the album covers to visualize a collection of albums *(see figure 2)*. The playlists view *(see figure 3)* is a bit more advanced though. It does not only list the current tracks of the playlist (or all the playlists available when using another view) but also appends some beneficial information to the whole view like other albums from the same artist or similar artists. It also provides automatical playlists by searching through the whole music collection (as defined by the user) and merging it using the tags chosen by the user. Track-wise visualizations are available by installing a plugin *(see figure 4)*

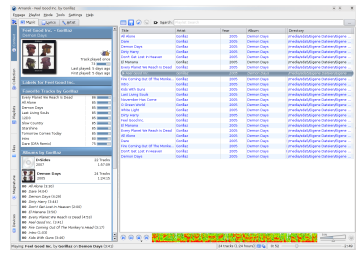

Fig. 3. Amarok Playlist

2.2.3 iTunes

With the spread of the iPod[3], a portable music (and nowadays also media) player developed by Apple, their associated software called iTunes has experienced an increasing distribution as well. Similar to the Windows Media Player it offers a grid-based overview of music albums included to the iTunes library *(see figure 2)* but also a so-called coverflow view that reminds of a jukebox *(see figure 5 bottom)*. Again, just as the Windows Media Player, they use animated art *(see figure 4)* to visualize tracks on their own, the only outstanding difference is the fact that they use 3D. With the integration of the newly developed feature called "Genius" (more information at section 3.2.3) they approach the research done in the MIR field.

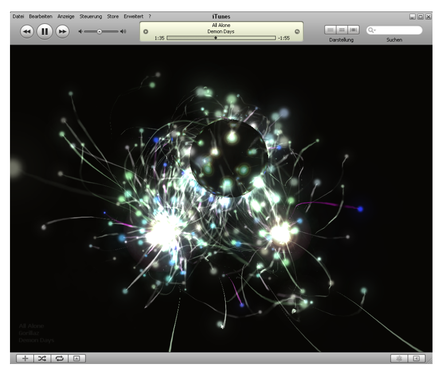

Fig. 4. Track-wise Visualization (example from iTunes)

3 Current 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" [14]. But to do this, you first have to identify what is actually relevant, and what is not. As explained before, music and listening to music is a subjective concept, so it is intuitive that human opinion should lead us the way on how to set up the automatism behind our systems. But as [2] stated, it is almost impossible to get human opinion into one consistent basis you could work

[2]regarding version 1.4

[3]market share of 70% - 80% due to [7]

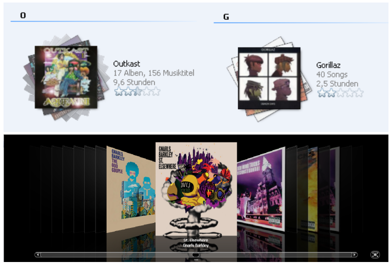

Fig. 5. Clustering (WMP, iTunes)

on. This is one of the main reasons why different research in the MIR field base on different approaches on how to set up their systems. And as no ground truth nor a common evaluation database exists [22] it is yet not possible to say which method works. Even though most publications contain evaluations they can hardly be compared as they only focus on their own approach. Basically when creating new systems there are three main issues to consider that influence the further work.

- What is the purpose of the system (use cases) (3.1)
- What method(s) will be used
- What visualization(s) be used

At the current point of research a couple of common use cases are known and supported by developed systems. Many researches follow the content-based query-by-example (3.1.1) but some also focus on different approaches (3.1.3). After the purpose has been clarified the next step is to decide which method will be used. The predominant method in most proceedings examined in this paper use methods of similarity measuring (3.2). The third issue to consider will then be the choice of the visualization (section 4).

3.1 Music Interaction & Description

This section implies common cases of interaction with music, ideas to improve this interaction, thereby created requirements for music players and different kinds of musical representation and description to fullfil those requirements. As the following will show there is a high relation between the interaction case and the used methods.

3.1.1 Query-By-Example (QBE)

As stated above, much recent research is based on the Query-By-Example paradigm. This means that one musical track or a part of it is used as an example to set up a query for relevant results. This intends to find similar music to the one used as an example. As we have heard before, listening to music is a very emotional and mood-based process. Therefore the choice of the music must be somehow related to our current mood. Since the user, at some point, cannot keep track anymore of a very large and still growing music collection, the search for music that fits special properties gets very difficult. By using known music as an example to find other similar music this process gets a lot easier and supports the user in his aim for music that he likes. But QBE is not only used to find music that a person would like to listen to at the moment. There are some more interesting use cases as following:

- Analyzing a composers evolution and his influences [30]
- Resolving copyright issues [30]
- Query-By-Humming (see 3.1.2)

3.1.2 Query-By-Humming (QBH)

Query-By-Humming is a special use case of Query-By-Example. QBH is used when a user only knows the melody or a tune from a song and hums it into a microphone. The input is then digitized and converted into a symbolic representation and compared with a database. Therefore QBH only works if a symbolic representation (3.2.1) is available for the matching song. "The conversion of generic audio signals to symbolic form, called polyphonic transcription, is still an open research problem in its infancy" [31]. Further research on QBH has been done by [12] [17] and others.

3.1.3 Query User Interfaces (QUI)

Contrary to the static Query-By-Example paradigm and related user interfaces, Tzanetakis et. al. [31] have developed and presented a couple of new user interfaces. Their goal was to support use cases that vary from the standard QBE and present user interfaces to suit them. While QBE methods base on a music file, a melody (see 3.1.2) or such as an input, and deliver data in context to this input, [31] investigate systems that rely on different data input. They define the term of **Query User Interfaces (QUI)** that sums up "any interface that can be used to specify in some way audio and musical aspects of the desired query". The first and simplest technique they present is the use of sound sliders to adjust the parameters for the desired query. While the QBE paradigm already relies on audio input, this technique has numerical input but auditory output. This is called sonification. So-called sound palettes are similar to the sound sliders, the main difference here is that they offer a fixed amount of values (a palette) for each attribute while the rest works just as with sound sliders. The 3D sound effect generator offers a possibility to query music by interaction, thus meaning that the system provides representation of physical objects (such as a can and different surfaces) the user can choose from and make them interact (roll the can on a wooden table). Therefore a use case based on actions or interactions as input is supported. Last but not least they present user interfaces based on midi input. With systems such as the groove box or style machines the user has the opportunity to use symbolic representations (in this case midi data) for a query to search large audio databases.

3.1.4 Query-By-Rhythm (QBR)

[6] propose another technique called Query-By-Rhythm where they support queries based on rhythmic similarity. At first they model a song into a rhythmic string used as a representation and thereby transform the similarity measuring into a string-based process. A rhythm string is created by measuring rhythm patterns (called mubol) and notes as well as their occurence time, ignoring their pitch value. By defining similarity for the mubols they are then able to educe similarity for rhythm strings from it.

3.2 Method

Music itself is generally self-similar [10] (see section 4.3). Together with the fact that similarity measuring is a common technique used to organise data in general it provides one of the basic methods used in MIR. Within music collections it is used to compare different songs and use the output to arrange matching data together and vice versa. This supports use cases like finding new music that still suits the user's taste and/or his current mood. There are different opinions on what is actually relevant to indicate similarity but they all have common principles behind them. Usually the thereby extracted information is then later on used to set up the borders for the visualization. There are three basic ways of measuring similarity. The first one is by looking at symbolic representations of the music such as written notes (3.2.1), another one is by measuring acoustic properties (3.2.2) and the last one is by measuring subjective characteristics (3.2.3).

3.2.1 Symbolic

Using symbolic representations is popular method to compare and extract similar music. It is even indispensable for some ideas such as 3.1.2 and 3.1.4 and often supported by systems that do not only rely on symbolic representation as it expands the possibilities. Symbolic representations usually appear as

- lyrics
- scores

- midi representations
- rhythmic patterns

and the like. Systems that use symbolic data to search through their database usually work with string-based methods, set-based methods or probabilistic matching [30]. With string-based methods it is usually important to grant some divergence as the desired result will not always be an exact match. Set-based methods do not have this problem as they donot rely on a set order. They work with ”properties like time, pitch, and duration” [30]. ”The aim of probabilistic matching methods is to determine probabilistic properties of candidate pieces and compare them with corresponding properties of queries” [30].

3.2.2 Acoustic

Acoustic based measuring is a technique that is opposite to metadata-based measuring and such, as it relies on data derived from the audio signal and not data appended to the audio. The basic idea behind acoustic measurement is to extract information from the raw audio input and use it to create a model representation. The relevance of the different possible attributes cannot be generalized as it depends on the requirements of the sytem and the subjective opinion of the developers. As we will see in section 5 researchers set up their own choice of relevant properties and measuring methods. The following lists a couple of commonly used attributes as described by [30]:

Loudness: can be approximated by the square root of the energy of the signal computed from the shorttime Fourier transform, in decibels.

Pitch: The Fourier transformation of a frame delivers a spectrum, from which a fundamental frequency can be computed with an approximate greatest common divisor algorithm.

Tone (brightness and bandwidth): Brightness is a measure of the higher-frequency content of the signal. Bandwidth can be computed as the magnitudeweighted average of the differences between the spectral components and the centroid of the shorttime Fourier transform. It is zero for a single sine wave, while ideal white noise has an infinite bandwidth.

Mel-filtered Cepstral Coefficients (often abbreviated as MFCCs) can be computed by applying a mel-spaced set of triangular filters to the short-time Fourier transform, followed by a discrete cosine transform. The word ”cepstrum” is a play on the word ”spectrum” and is meant to convey that it is a transformation of the spectrum into something that better describes the sound characteristics as they are perceived by a human listener. A mel is a unit of measure for the perceived pitch of a tone. The human ear is sensitive to linear changes in frequency below 1000 Hz and logarithmic changes above. Melfiltering is a scaling of frequency that takes this fact into account.

Derivatives: Since the dynamic behaviour of sound is important, it can be helpful to calculate the instantaneous derivative (time differences) for all of the features above.

This does not claim to be a universally valid list, just some basic possibilites. There will be further information in section 5 on what is actually measured in what research project.

3.2.3 Subjective

As we have heard before, music is subjective. People judge and categorise music differently, based on their taste and their mood. As symbolic and acoustic measuring does not take this into account there is the technique of measuring subjective properties. The basic idea behind this is to analyze people’s behaviour and draw logical conclusions from it. A relatively new example for such is the Apple Genius[4] introduced with the 8th version of Apple’s iTunes (see 2.2.3). After searching the users music library and the Apple store it provides a playlist of up to 25 similar songs based on the user’s initial choice given to Genius. The similarity measuring uses analysis of people’s listening behaviour with iTunes amongst other things. Nowadays large communities of music-interested users, providing statistical data of their listening behaviour exist such as Pandora [5], Last.fm [6], Playlist [7] and Imeem [8]. Based on the user-provided data they create playlist for different tastes, moods or events and offer recommendations to the user. This is achieved by collaborative filtering of all the user data assuming with the input from one user assuming that the predictions will then fit his musical taste. Collaborativ filtering means to create links by measuring time-near apperance of artists, albums or tracks in users playlists, numerical apperance in charts and the total occurence of that item within the whole community as well as other features. Even though subjective means are indispensable to measure cultural and other intangible factors it can only be used if a large amount of associated data already exists so it cannot be applied to new or unknown artists and music. Subjective similarity measuring has been used and analyzed on several occasions (see [26] [11] [2])

4 VISUALIZATION

Nowadays there exists an enourmous amount of different visualization techniques. This section will list just a few of them (as proposed by [26]) that relate to visualizing music and music collections.

4.1 Similarity

The following describes a few visualization examples used to describe similarity (for example measured as described in section 3.2) - mainly between artists but it could also be used with other attributes.

Self-Organizing Map (SOM) The Self-Organazing Map ([15] [16]) is a neural-network algorithm to organize data with multi-dimensional feature vectors and map them into a non-linear, low-dimensional (usually 2D) visual representation. The discreet output space is split into so called *map units* from which the best matching unit is chosen for each item to represent it. It tries to map close data from the feature space (thus meaning a high similarity) to close units in the output space and thereby clustering the data. The SOM algorithm is an extremly popular method (over 5000 scientific articles that use it according to [15]) that is also used by some of the examples presented in section 5.

Multi-Dimensional Scaling (MDS) The basic aim behind multi-dimensional scaling is to maintain an appoximate representation of distance from the data itself to its visualization. This means to represent the distance between two data objects by their attributes (similarity in terms of artist, genre and so on) as good as possible. This is usually done by firstly assigning a random position in the output space and then re-arranging the objects by calculating new coordinates to still stick to the given distances and minimize the error rate. Research has developed a couple of different algorithms trying to fulfill the strict specifications of MDS such as *Sammons mapping* [25]. *Figure 6* shows an example of multi-dimensional scaling as presented in [26].

Continuous Similarity Ring (CSR) The Continious Similarity Ring is a novel visualization technique developed by [26] (*see figure 7*). It is based on similarity measuring with anchor references (one prototype artist) for each genre arranged as a circle. Artists similar to the genre prototype are then appended to it while the

[4] Additional information at `http://www.apple.com/de/itunes` December 2008
[5] `www.pandora.com` December 2008
[6] `www.last.fm` December 2008
[7] `www.playlist.com` December 2008
[8] `www.imeem.com` December 2008

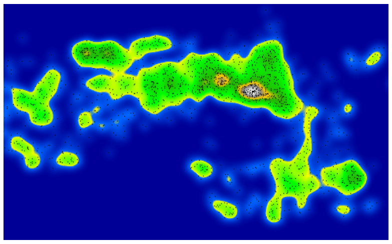

Fig. 6. Multi-Dimensional Scaling Example by [26]

arrangement of the prototype artist tries to preserve the similarity as a distance representation. A thick and colourful edge as well as a close distance between two nodes means a high similarity whilst thin and dark edges connect less similar artists. [26] uses Johann Sebastian Bach and Willie Nelsons as an example to show the functioning (no other prototypes connected to the classic genre - Folk and Country closely connected). He refers the malfunctioning part, as can be seen with the artist Bush, to problems with the measuring algorithm that only occur when using a small amount of data.

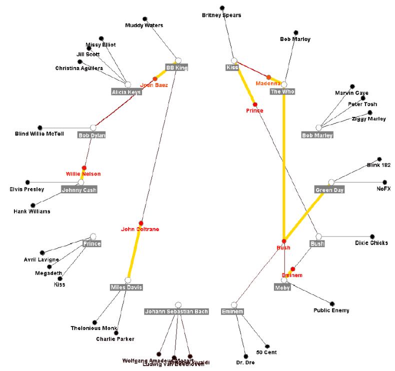

Fig. 7. Continuous Similarity Ring (CSR) by [26]

4.2 Hierarchical

In contrast to the techniques presented in section 4.1 this section will feature techniques used to visualize hierarchical structures.

Treemap Treemaps are a quite common and popular method to visualize hierarchical data. The available space in form of a rectangular layout is recursively divided into further rectangles to indicate the underlying steps in the hierarchy and filled with the associated items of the data set. Even though it is a well-known visualization technique and several further developments have been made ([3] [32]) it is no common technique used for music and music collections.

Hyperbolic Tree The Hyperbolic Tree's original name was Hyperbolic Browser, but due to its tree-like data structure and visual form it is also referred to as a tree [26]. As the name already suggests the tree is is laid out in a hyperbolic plane thus growing with the size of its radius and providing large space at its outer border. Each node of the tree is assigned an equal share of the total angle to arrange its successors. Each child of a node is placed around the arc of the node's angular share and therefore has the same distance. By doing this a possible overlap of children is prevented. The root node is then used as the center item while the other items are arranged accordingly - the further from the center, the deeper the item is in the hierarchy. *Figure 8* shows examples for a hypertree taken from the Sonic Browser (section 5.3) and Musictrails [9]. The Musictrails example shows a good result putting Damon Albarn, lead singer of the Gorillaz, just next to his band.

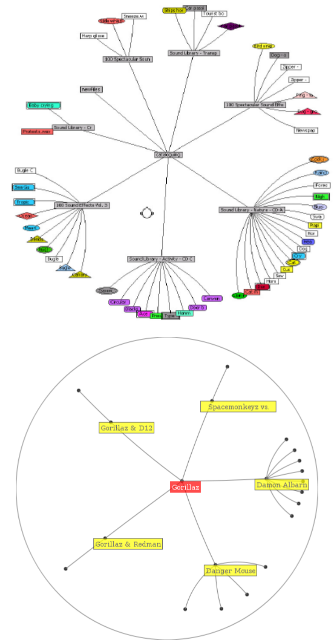

Fig. 8. Hypertree: Sonic Browser (section 5.3) & Musictrails

Sunburst (InterRing) The Sunburst or InterRing (*see figure 9*) is a circular visualization concept developed by [1] and [29]. The top-of-the-hierarchy item is in the center while each sub-level is presented by an arc where the distance to the center again indicates the depth. All children of a node are bounded to the same amount of angular extent their parent had and drawn within those borders. The size of a node is calculated by its proportion to the remaining nodes at the same level. A major disadvantage of the Sunburst visualization is that elements at the bottom of the hierarchy get assigned a very small deal of the arc and therefore are very hard to spot - a problem growing straight proportional with the data size.

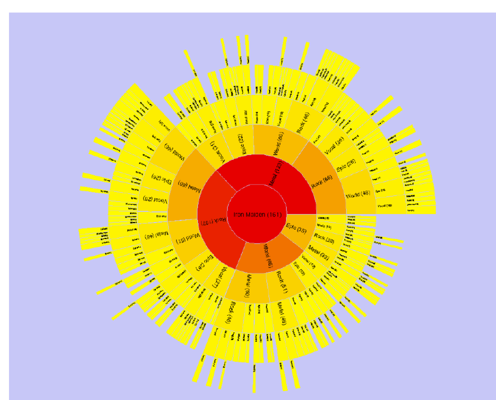

Fig. 9. Sunburst by [1]

Stacked Three-Dimensional Sunburst The Stacked Three-Dimensional Sunburst has been developed by [26] and operates as an extension to the Sunburst concept. The main motivation were the short-comings of the original system which only provided for two-dimensional data to be handled. The support of multi-dimensions is handled by adding the height as a new scale

[9] `http://www.musictrails.com.ar` December 2008

thus making the visualization 3D. With this feature added it is possible to have one layer each to display every data dimension. To prevent an infinite growth the authors introduced a number of limitations for the number of nodes, the depth of the whole stack and the minimal angular extent per node which solves the second deficit of the original system. *Figure 10* shows a non-labeled example of the Stacked Three-Dimensional Sunburst system with three layers where color is used to distinguish the data dimension represented by the arc's angle.

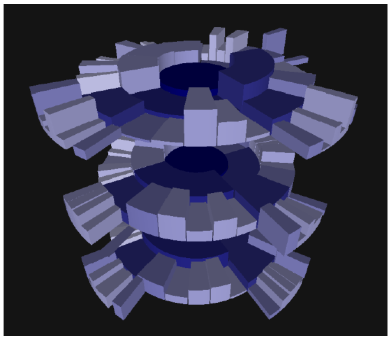

Fig. 10. Stacked Three-Dimensional Sunbursts developed by [26]

4.3 Track-wise

Whilst visualizing music collections has experienced a lot of attention the research field of visualizing single tracks has often been left behind and did not receive much attention. The first attempts to visualize single tracks were done by using timebars which did not only give a visual information but already allowed intra-track navigation [33]. The common strategy to visualize songs nowadays is by using dynamic art depending on the acoustic properties (*see figure 4*). [33] developed a new visualization to aid intra-track navigation called *moodbar* (*see figure 12*) that has already been included in music players like Amarok. The moodbar is a visual track representation that indicates different parts of the song by using different colours. It uses several measuring techniques to extract one and three-dimensional information where the first is used to set the luminosity of a grey shade while the second values are formatted into a RGB colour. Therefore similar clips of one track obtain corresponding coloring and thereby indicate their likeness. [10] developed the concept of self-similarity to visualize the time structure of a musical piece. By comparing it to itself he creates a 2D representation with two time scales where the brightness of each square in the graph represents the similarity. A high similarity is bright while dissimilarity is dark. Because of the identic scales there is a bright diagonal from bottom left to the top right (*see figure 11*). This helps to identify different sections (like verses and chorus) of the song. Unfortunately visualizing tracks seems to be not as popular even though it would provide a lot of possibilities for future research.

5 Related Work

This section will introduce a selection of the mentioned research, shortly explain the methods used to retreive the information and the chosen type of visualization.

5.1 AudioRadar

AudioRadar is a system developed by [19] at the University of Munich. As the name already foreshadows it relates to the metaphor of a ships' radar. The system is intended to aid QBE (see section 3.1.1) use cases and uses acoustic measuring (see section 3.2.2) to calculate the distance between two songs and arrange them accordingly. The example object is in the center (*see figure 13*), appended with some visual controlling options, while the other items are represented by small dots. The system integrates four axes whereby the most dominant difference from the example is chosen to determine the axis that gives the direction. The size of the distance sums up the similarity or difference.

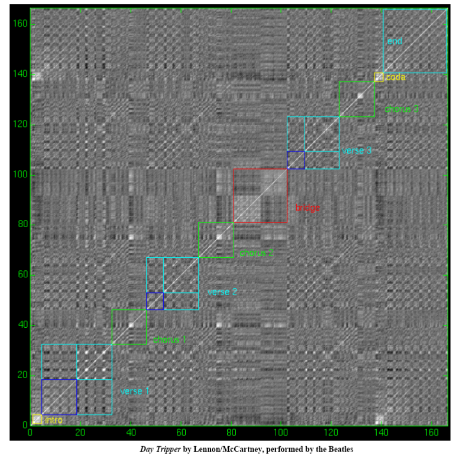

Fig. 11. Self Similarity example by [10]

Fig. 12. Amarok using the moodbar plugin by [33]

The user has the possibility to re-arrange the whole by chosing a new center. The attributes extracted with the automatical acoustic analysis were

- slow vs. fast
- clean vs. rough
- calm vs. turbulent
- melodic vs. rhythmic

by using a given analysis library. The two-dimensional projection of those four attributes is done by picking two of the scales to map them on the four axes. AudioRadar also provides the creation of mood-based playlists by giving the user the oppurtunity to define the range of attributes for the chosen music (similar to the sliders from [31] in section 3.1.3).

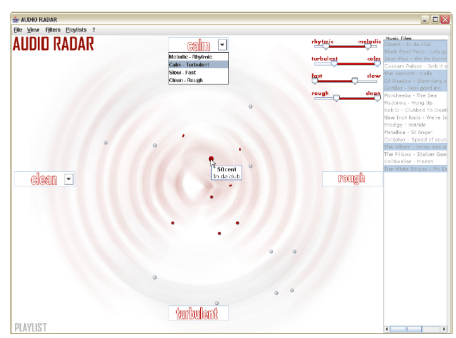

Fig. 13. Audio Radar by [19]

5.2 AudioPhield

AudioPhield is a novel multi-touch tabletop display system created by [27] to support multiple users. It allows for multiple users to interact with each others music collection and easily spot concuring areas. As the whole visual layout is again based on similarity metrics (close means similar, far away means different) such areas are dedicated to a certain kind of music and so a high conformity means a similar musical taste. Each song is represented as a dot and, of course, every user's music collection is assigned a different color to avoid confusion (*see figure 14*). They use a mixture of acoustic and subjective methods to measure similarity and attach meta information about the accuracy to each value. The visualization is based on a SOM with 80 nodes per axis to enable "fine nuances in close proximities" but still "make searches and modifications computable in reasonable time" [27]. Unlike within normal SOM training methods AudioPhield does not recompile the best matching unit but only does it once. This beholds the risk of units "moving away" from an item which is prevented by preimprinting the SOM to define a rough layout. To avoid overlapping of single items they integrated a so-called "*spring-algorithm*" that makes the items push off from each other while they are still connected to their inital place and thereby rearrange themselves until they do not overlap anymore.

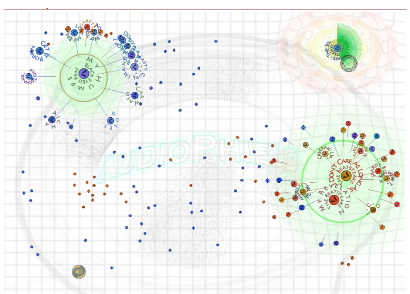

Fig. 14. AudioPhield by [27]

5.3 Sonic Browser

The Sonic Browser has been developed by Eoin Brazil and Mikael Fernstroem and has lived trough several enhancements ([8] [9] [4] [5]). Its main intent is to provide aid to humans for browsing through large audio collections and exploring them. They provide a couple of different views to denote the relationship within the data. The focus lies on the presentation of the content rather than on the classification. The foundation of the design for the Sonic Browser are the "principles of direct manipulation and interactive visualisation interfaces proposed by Shneiderman [28]. The three primary facets of this foundation are "overview first, zoom and filter, then details on demand" [4]. The Sonic Browser integrates many different views *(see figure 15)* such as a basic x-y plot (first implementation), the Starfield Display and a Hypertree View (*figure 8*). The Starfield Display is a scatterplot, with axis depending on the attributes of the dataset, which supports axis remapping (to other attributes), drag & drop as well as object-selection and -editing. Information on the Hypertree can be obtained from [13] & section 4 while a short explanation on the TouchGraph is available at [4]. What is special about the Sonic Browser is the fact that it maps attributes from the data to the representation of an object by adjusting its shape. For example changing symbol size to represent the file size, colour for the sampling rates, symbol shape for the file type and location for the date and time as used in the Starfield Display.

5.4 Islands Of Music (IOM)

Islands Of Music[10] was developed by Elias Pampalk ([21], [23]) as his master thesis [20]. He uses acoustic measuring by dividing a song into time intervals of 6 seconds each and analyzing every third one further. After calculating the loudness sensation per frequency band in a 12ms time interval and then analyzing the loudness modulation in the whole time interval by using a Fourier Transformation as well as several following steps (further detail in [21]) he calculates a median of all sequences (first and last one is cut to avoid fade-in and fade-out effects) to represent the musical piece. Evaluation of alternative combination methods figured for the median to yield likewise results. The

[10] http://www.ofai.at/~elias.pampalk/music/

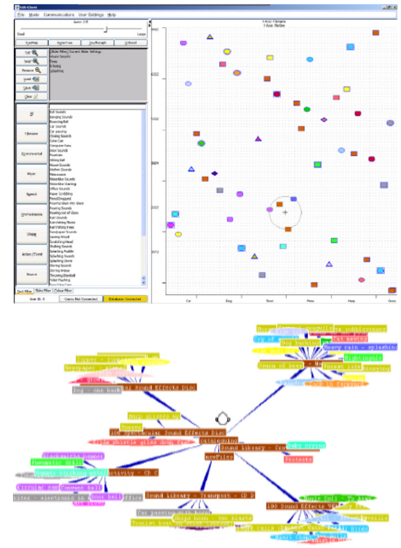

Fig. 15. Sonic Browser Views: Starfield Display & Touchgraph by ([8] [9] [4] [5])

novel user interface represents a music collection as a series of islands (*see figure 16*), each representing a musical genre, with labeled hills and mountains to describe rhythmic properties. Similar genres are locacated close to each other just as the respective tracks on the islands are. Again the arrangement of the music collection is based on similarity metrics while the visualization works with a Self-Organizing Map trained with the received data. [21] developed a new technique that allows each track to vote for a unit that best represents it - giving it one, two or three points - and uses the achieved ranking to set up the geographic maps.

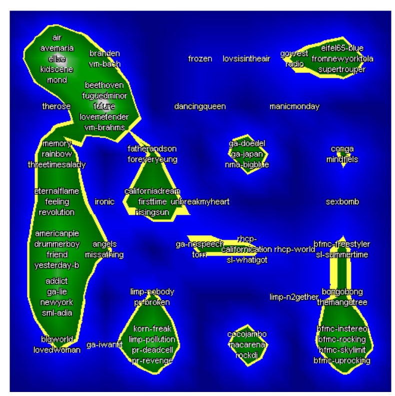

Fig. 16. Islands of Music Example using a 7x7 SOM on 77 songs

5.5 3D User Interface by [14]

The 3D User Interface developed by [14] is an innovative virtual reality like system to browse and explore music collections. The measuring is done with "rhythm-based Fluctuation Patterns that model the periodicity of a audio signal" as presented by [21] and explained in section 5.4. The visualization follows *Islands Of Music* [21] concept but adds additional functionality. While the original layout was in 2D [14] present a 3D approach by using the result from the SOM and feeding it with a Smoothed Data Histogram (SDH) interpreted as a height value to generate the third dimension. The height of the landscape corresponds to the amount of similar items in the specific region thus meaning a high mountain acts for many songs. By creating this virtual reality the user is invited to perform an interactive tour through his music collection which does not only integrate visual information but also auditory. When browsing through the collection audio thumbnails of close songs are played to further strenghten the impression. [14] also included a web-based information retrieval for textual or graphical (covers) information that is appended to the representation *see figure 17*. In total they provide a system based on known ideas ([21]) and enhance it with a new dimension as well as further functionality to support the user.

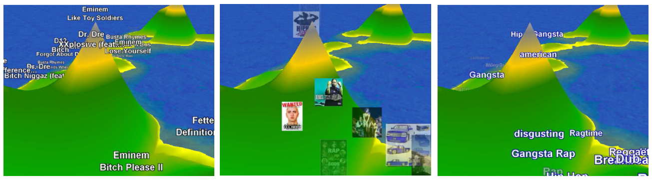

Fig. 17. 3D User Interface by [14]

6 Conclusion And Future Outlook

The present survey has demonstrated that development in Music Information Retrieval & Visualization has happened and is even partially included into modern music tools already. Most research has a common basis relating to ideas and methods such as using similarity measuring to model a music collection which are than used to create appealing interfaces for the user. As seen within section 5.5 some techniques even rely on previous work which indicates the floating development. Even though the basic principles usually stay the same the exact implementation (chosen method and measured properties) and results differ which can be traced back to individual opinion as well as a missing proven concept for "the" best-working method. Another problem that adds on constraining the comparing of various developments is the missing of a common dataset to evaluate invented techniques. As almost each developer relies on his own dataset it is barely a miracle that the results vary a lot. For the future a further consideration of track-wise and intra-track visualization will be a possible new issue. With growing social music societies such as last.fm the future focus of measuring similarity will probably lie within subjective means as those platforms provide a large basis.

References

[1] K. Andrews and H. Heidegger. Information slices: Visualising and exploring large hierarchies using cascading, semi-circular discs. In *Proc of IEEE Infovis 98 late breaking Hot Topics*, pages 9–11, 1998.

[2] A. Berenzweig, B. Logan, D. Ellis, and B. Whitman. A Large-Scale Evaluation of Acoustic and Subjective Music-Similarity Measures. *Computer Music Journal*, 28(2):63–76, 2004.

[3] T. Bladh, D. Carr, and J. Scholl. Extending Tree-Maps to Three Dimensions: A Comparative Study. *LECTURE NOTES IN COMPUTER SCIENCE.*, pages 50–59, 2004.

[4] E. Brazil and M. Fernstrom. Audio information browsing with the Sonic Browser. In *Coordinated and Multiple Views in Exploratory Visualization, 2003. Proceedings. International Conference on*, pages 26–31, 2003.

[5] E. Brazil, M. Fernstrom, G. Tzanetakis, and P. Cook. Enhancing Sonic Browsing Using Audio Information Retrieval. In *International Conference on Auditory Display ICAD-02, Kyoto, Japan*, 2002.

[6] J. Chen and A. Chen. Query by rhythm: an approach for song retrieval in music databases. In *Research Issues In Data Engineering, 1998. Continuous-Media Databases and Applications. Proceedings. Eighth International Workshop on*, pages 139–146, 1998.

[7] P. Elmer-DeWitt. How to grow the ipod as the mp3 player market shrinks. *CNN Money*, 2008.

[8] M. Fernström and L. Bannon. Explorations in Sonic Browsing. In *Proceedings of HCI on People and Computers XII table of contents*, pages 117–131. Springer-Verlag London, UK, 1997.

[9] M. Fernstrom and E. Brazil. Sonic Browsing: an auditory tool for multimedia asset management. In *Proc. Int. Conf. on Auditory Display (ICAD)*, 2001.

[10] J. Foote. Visualizing music and audio using self-similarity. In *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, pages 77–80. ACM Press New York, NY, USA, 1999.

[11] G. Geleijnse and J. Korst. Tagging artists using cooccurrences on the web. In *Proceedings Third Philips Symposium on Intelligent Algorithms (SOIA 2006)*, pages 171–182.

[12] A. Ghias, J. Logan, D. Chamberlin, and B. Smith. Query by humming: musical information retrieval in an audio database. In *Proceedings of the third ACM international conference on Multimedia*, pages 231–236. ACM New York, NY, USA, 1995.

[13] B. Johnson and B. Shneiderman. Tree-Maps: a space-filling approach to the visualization of hierarchical information structures. In *Proceedings of the 2nd conference on Visualization'91*, pages 284–291. IEEE Computer Society Press Los Alamitos, CA, USA, 1991.

[14] P. Knees, M. Schedl, T. Pohle, and G. Widmer. An Innovative Three-Dimensional User Interface for Exploring Music Collections Enriched with Meta-Information from the Web. In *Proceedings of the ACM Multimedia*, pages 17–24, 2006.

[15] T. Kohonen. The self-organizing map. *Neurocomputing*, 21(1-3):1–6, 1998.

[16] T. Kohonen. *Self-Organizing Maps*. Springer, 2001.

[17] L. Lu, H. You, and H. Zhang. A new approach to query by humming in music retrieval. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, 2001.

[18] K. J. O'Brien. As eu debated, microsoft took market share. *International Herald Tribune*, 2007.

[19] R. K. A. B. Otmar Hilliges, Phillipp Holzer. AudioRadar: A metaphorical visualization for the navigation of large music collections. In *Proceedings of the International Symposium on Smart Graphics 2006, Vancouver Canada*, July 2006.

[20] E. Pampalk. Islands of music: Analysis, organization, and visualization of music archives. Master's thesis, Technischen Universit, Vienna, Austria, 2001. Diplomarbeit.

[21] E. Pampalk. Islands of Music Analysis, Organization, and Visualization of Music Archives. *Journal of the Austrian Soc. for Artificial Intelligence*, 22(4):20–23, 2003.

[22] E. Pampalk, S. Dixon, and G. Widmer. On the evaluation of perceptual similarity measures for music. In *Proceedings of the International Conference on Digital Audio Effects (DAFx-03)*, pages 7–12, 2003.

[23] E. Pampalk, S. Dixon, and G. Widmer. Exploring Music Collections by Browsing Different Views. *Computer Music Journal*, 28(2):49–62, 2004.

[24] Z. Pousman, J. Stasko, and M. Mateas. Casual Information Visualization: Depictions of Data in Everyday Life. *IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS*, pages 1145–1152, 2007.

[25] J. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, 18(5):401–409, 1969.

[26] M. Schedl. *Automatically Extracting, Analyzing, and Visualizing Information on Music Artists from the World Wide Web*. PhD thesis, Johannes Kepler University, Linz, Austria, July 2008.

[27] M. Schicker. Audiophield: Exploring casual collaborative browsing of large music collections. Master's thesis, Ludwig Maximillian Universitaet, Munich, Germany, 2008. Diplomarbeit.

[28] B. Shneiderman. Designing the user interface strategies for effective human-computer interaction. *ACM SIGBIO Newsletter*, 9(1), 1987.

[29] J. Stasko and E. Zhang. Focus+ Context Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualizations. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 57–68, 2000.

[30] R. Typke, F. Wiering, and R. Veltkamp. A Survey of Music Information Retrieval Systems. In *Proceedings of the International Conference on Music Information Retrieval*, pages 153–160, 2005.

[31] G. Tzanetakis, A. Ermolinskyi, and P. Cook. Beyond the Query-by-Example Paradigm: New Query Interfaces for Music Information Retrieval. In *Proceedings of the 2002 International Computer Music Conference*, pages 177–183, 2002.

[32] J. van Wijk and H. van de Wetering. Cushion Treemaps: Visualization of Hierarchical Information. In *Proceedings of the 1999 IEEE Symposium on Information Visualization*, page 73. IEEE Computer Society Washington, DC, USA, 1999.

[33] G. Wood and S. OKeefe. On Techniques for Content-Based Visual Annotation to Aid Intra-Track Music Navigation. In *Proceedings of ISMIR 2005, 6th International Conference on Music Information Retrieval*, pages 58–65, 2005.