# An Explorative, Hierarchical User Interface to Structured Music Repositories

Markus Schedl

December 2003

## Abstract

Due to efficient compression algorithms like MP3, the number and size of digital music repositories have increased dramatically over the past few years. Consequently, the demand for obtaining digital music from the Internet has been rising. Hence, effective methods for finding pieces of music in such repositories are becoming more and more important. Unfortunately, when working with traditional user interfaces which solely provide text-based search, the user already has to know certain textual properties of the songs he/she is looking for (e.g. name of the artist or album). In contrast, the prototype of the user interface which has been developed by the author is based on graphical visualizations of musical similarities between the pieces contained in the repository. This enables the user to exploratively browse through the collection, an approach which is especially useful for discovering formerly unknown pieces of music.

In order to provide different views of the music collection, two algorithms were chosen to process the audio signals. These algorithms measure musical similarities according to rhythmic and timbral properties.

The developed user interface *"ViSMuC" (Visualization of Structured Music Collections)* implements an artificial intelligence method called *Aligned Self-Organizing Maps* in which high-dimensional data is represented by a 2-dimensional map. The pieces of music are visualized according to an adjustable weighting of their rhythmic and timbral properties. Forming clusters of similar pieces, the resulting groups are colored with respect to the number of songs they represent. Different colormaps are available for this purpose. Since illustrating all pieces of a medium or large collection on a single map would yield a tremendously complex and thus unusable visualization, the user interface contains two hierarchical components. Firstly, for each region of the map that represents a large number of songs, a new map is provided. Secondly, the directory structure of the repository is taken into account since it usually forms a meaningful hierarchy. Another important part of the user interface is the visualization of arbitrary meta-information, which can be taken, for example, from ID3-attributes or external databases. The employed technique illustrates the distribution of the values assigned to the meta-information attributes over the complete map. Together with visualizations that are based on the features gained from the similarity measures and their projection to the map, the images showing these distributions facilitate the interpretation of the map.

For the purpose of testing the user interface, a test repository composed of more than 800 MP3-files was created and various meta-data was inserted into a database by the author. Finally, a short usability study was conducted and suggestions for applications as well as for improvement of the prototype were elaborated.

## Motivation and Introduction

Over the past few years, the demand for digitally stored music has risen drastically. The availability of algorithms for music compression, especially MPEG-Layer 3 (MP3), together with high-speed Internet access, yielded an enormous increase in digital music distribution (DMD). The growing number of large music databases, which are very important for commercial music stores like *AMG All Music Guide* [1], *Amazon* [2] or *iTunes* [3], just to name a few, raises the demand for methods to efficiently browse through and search in such repositories. Most of the existing interfaces perform quite well when the task is to find music by a given artist or on a specified album, i.e. when the user knows exactly what he/she is looking for, but are unsuitable to support the user in discovering unknown music. For this reason, a user interface based on *Self-Organizing Maps (SOMs)* – neural networks used to cluster high-dimensional data – has been developed. The data consist

[1] *http://www.allmusic.com*
[2] *http://www.amazon.at*
[3] *http://www.apple.com/itunes*

of feature vectors, each of which describes some musical properties, e.g. rhythm or timbre, of one piece of the repository.

The basic idea of the user interface is to visualize the repository on a map where songs that are similar according to a certain property can be found close together. Thus, there are regions representing a lot of pieces as well as very sparse areas on the map. These differences in density can be visualized by applying a colormap, which enables the user to distinguish certain clusters, each of which represents music with similar rhythmic or timbral properties. In the developed interface, a colormap similar to the *"Islands of Music"* described in [Pam01] is used by default, because the metaphor of geographic maps where the clusters are represented by islands which are separated by the sea seemed to be very intuitive to the user. In addition, other colormaps are available.

The user interface is based on the visualization approach of [Pam01]. This original approach restricts the visualization to one single map, thus is not suitable to deal with large music repositories. Addressing this shortcoming, the developed prototype combines different hierarchies, which are not necessarily compatible, and visualizes them by linked maps. Unlike in [Pam01], two existing similarity measures were chosen to let the user shift the view between a rhythmical and a timbral clustering of the repository. The choice for the two measures is based on an analysis of five similarity measures.

Furthermore, the developed system also takes into account the hierarchical structure of the music repository, which comprises two aspects – the musical structure and the directory structure.

The former is given by the distribution of the pieces on the map. If the number of songs assigned to one region exceeds a fixed limit, a further refinement of this area is done by introducing a new hierarchical level, i.e. displaying a new map which contains only the pieces of the particular region. In this case, just a prototype piece that best represents the music of the underlying hierarchy level is displayed on the original map. Using this technique of hierarchically organizing the map according to the musical similarity structure of the pieces, the system is capable of visualizing an unlimited number of tracks.

The latter, the directory structure, is usually very important to the provider of music databases since it offers an easy way to organize the repository on the harddisk according to varying preferences. For example, a supplier of digital music could create directories for each genre, whereas another may prefer to name them after the artists. For this reason, the user interface offers the possibility to easily jump into the directory of each displayed piece of music.

Moreover, the user can get deeper insights into the clustering of the pieces by browsing different views. This is done by shifting the focus between timbral and rhythmic aspects, which leads to different maps and clusters [PDW03].

Another important part of the user interface is the visualization of additional meta-information. Since many people use the *ID3 tagging system* [4] to label and categorize their songs, presenting these data is usually very valuable to the user. Therefore, they are visualized in two ways. Firstly, a textual presentation of the most frequently used ID3-attributes appears when the mouse is moved over the label of an arbitrary song. Secondly, a graphical visualization of the genre distribution is shown for each map, which supports the user in interpreting the clusters, i.e. assigning a genre to each cluster.

Due to the fact that the ID3-standard defines a limited number of attributes, one could prefer using a database providing advanced information for each track of the repository. The user interface also permits the visualization of such extra information given by an external database.

On the whole, the developed user interface offers a wide range of possibilities to exploratively discover formerly unknown music as well as to browse through well-known repositories. It can be used by commercial music stores that want to offer an explorative way of finding new music according to the personal taste of their customers. Private music lovers owning large databases could also be in favor of such a system.

## Feature Extraction

To calculate the similarity of two arbitrary pieces of music, it is necessary to extract low-level features from the audio signal. These features describe specific musical properties like timbre or rhythm. For the calculation of the SOMs for the user interface, two techniques for feature extraction and similarity measurement were used.

The first one is a rhythm-based measure called rhythm patterns/modified fluctuation strength (RP/MFS). It aims at detecting reoccuring beats at certain frequencies. For this reason, a matrix describing the modified fluctuation strengths for 20 frequency bands and 60 levels of modulation frequencies ranging from 0 to 10 Hz, thus from 0 to 600 *beats per minute (bpm)*, is calculated for every piece of music. Technical details of the algorithm can be found in [PRM02a].

As second feature extraction technique, spectrum histograms (SH) are used. Such spectrum histograms describe timbral aspects of music. The approach presented in [PDW03] and used here is quite simple and very

[4] *http://www.id3.org*

fast compared to other timbre-based techniques, but nevertheless yields respectable results. The calculation of a spectrum histogram matrix for a specific piece of music is done by counting how many times the piece of music reaches or exceeds a specific loudness in each frequency band. Hereafter, the sum of the resulting matrix is normalized to 1 in order to cope with the different play lengths of the pieces in a collection. The resulting matrix has 20 rows and 50 columns. The rows represent the frequency bands, whereas the columns indicate the loudness level.

# Organization and Visualization of High-Dimensional Data

The result of the feature extraction is a high-dimensional data item for each piece of music in the collection under consideration (1 200 dimensions in case of RP/MFS, 1 000 in case of SH). Since calculating the SOMs for the developed user interface directly from the original feature vectors would lead to unacceptable computation times, a dimensionality reduction to 80 is performed by compressing the data using *Principal Component Analysis (PCA)* [Hot33, Jol86]. The compressed feature vectors are then clustered with a *Self-Organizing Map (SOM).* Finally, the calculated SOM is visualized with a *Smoothed Data Histogram (SDH).* Since the SOM and the SDH are the most important techniques involved in the creation of the user interface, they are explained in the following.

### Self-Organizing Map (SOM)

The *Self-Organizing Map (SOM)* [Koh82, Koh01, SJ02, Ves00] is a powerful neural network algorithm based on unsupervised learning. The main idea of the SOM is to organize multivariate data on a usually 2-dimensional map in such a way that data items which are similar in the high-dimensional data space are projected to locations which are close to each other on the map. Therefore, probably the most important application area of the SOM is the representation of high-dimensional data sets.

Basically, the SOM consists of an ordered set of map units, each of which is assigned a *model vector*[5] $\mathbf{m}_i$ of the same dimensionality as the original data space. The map units are arranged either rectangularly or hexagonally to form a grid. The set of all model vectors of a certain SOM is referred to as its *codebook*.

Before training the SOM, the model vectors are initialized. This can be accomplished by assigning random values or by using more sophisticated methods. For example, the first $m$ principal components can be calculated in order to linearly initialize the model vectors along the $m$ greatest eigenvectors, where $m$ denotes the cardinality of the codebook, i.e. the number of map units.

The training process itself can either be performed sequentially or by using the *batch map* [Ves00]. Both of these methods are explained briefly below.

**Sequential Training** The basic algorithm for the SOM [Koh82] uses sequential training, also known as online training, which is performed iteratively. Each training iteration starts by choosing one randomly selected data item $\mathbf{x}$ out of the data set denoted by $\mathbf{X}$. Subsequently, the distance between $\mathbf{x}$ and each model vector $\mathbf{m}_i$ is calculated – e.g. according to the Euclidean norm. The map unit possessing the model vector $\mathbf{m}_{bmu}$ that is closest to the data item $\mathbf{x}$ is referred to as *best matching unit (BMU)* and is further used to represent $\mathbf{x}$ on the map. Formally, the selection of the BMU is given by Expression 1.

$$\|\mathbf{x} - \mathbf{m}_{bmu}\| = \min_i \{\|\mathbf{x} - \mathbf{m}_i\|\} \tag{1}$$

In the next step, the model vectors are updated to reduce the distance between the data item $\mathbf{x}$ and the model vectors of the BMU and its surrounding units. Since an important aspect of the SOM is to preserve the distances between the items in the data space, a neighborhood kernel $h_{bmu,i}(t)$ centered on the BMU is defined. Hence, the model vectors of units close to the BMU are adapted more than those far away from the BMU, which ensures that neighboring map units represent similar data items. This is of particular importance since, especially at the beginning of the training process and when random initialization is used, the model vectors exceedingly differ from the data items. The neighborhood kernel can be defined by a Gaussian as shown in Expression 2, where $\mathbf{r}_{bmu}$ and $\mathbf{r}_i$ denote the 2-dimensional position of the respective units on the map. Thus, by $\|\mathbf{r}_{bmu} - \mathbf{r}_i\|$ the distance between the units $bmu$ and $i$ within the output space is given. The time-varying parameter $\delta$ ensures a decreasing size of the neighborhood kernel during the training process, which enables the formation of large clusters in the beginning as well as allowing a selective fine-tuning towards the end of the training.

[5] In some publications the model vectors are denoted as *reference vectors*, *prototype vectors* or *weight vectors*.

$$h_{bmu,i}(t) = \exp\left(-\frac{\|\mathbf{r}_{bmu} - \mathbf{r}_i\|}{2 \cdot \delta(t)^2}\right) \tag{2}$$

Furthermore, a learning rate $\alpha(t)$ is used to gradually decrease the overall amount of adaptation. The complete update rule for the model vectors is given by Expression 3.

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha(t) \cdot h_{bmu,i}(t) \cdot [\mathbf{x} - \mathbf{m}_i(t)] \tag{3}$$

Either the learning rate and the neighborhood kernel decrease gradually with the iteration cycle $t$ since high adaptations are necessary in the rough training phase at the beginning and smaller ones for the fine-tuning towards the end of the training.

The number of performed iterations mainly depends on the cardinality of the training set and the number of map units but should be at least one epoch, i.e. each data item is presented once to the map.

**Batch Map** The batch map version of the SOM-algorithm as proposed in [Koh92] is also performed iteratively, but instead of presenting a single data item to the SOM at a time, the whole data set is taken into account at each iteration step. The main advantage in comparison with the sequential training is that executing the batch map algorithm on the same data set with the same parameters more than once produces similar maps. However, it is necessary to have access to the complete data set in order to use the batch map. Since all data items are presented to the map at the same time, no learning rate is needed.

Each training iteration involves two steps, which are executed until no further significant changes of the model vectors occur. First, the data set is divided according to the *Voronoi regions* of the map, i.e. the BMU for each data item is calculated (cf. Expression 1) and each map unit $i$ is assigned a *Voronoi set* $\mathbf{V}_i$ which points to all data items that are best represented by this unit. Having determined the Voronoi sets, in the second step, the new model vectors are calculated according to Expression 4, where $n$ denotes the number of items in the data set and $bmu_j$ is the best matching unit of data item $\mathbf{x}_j$. Therefore, the new model vector is the weighted average of the data items, where the weight of each item $\mathbf{x}_j$ is given by the value of the neighborhood function $h_{bmu_j,i}(t)$ at its BMU. Hence, the Voronoi sets which are spatially close to map unit $i$ influence the model vector $\mathbf{m}_i$ more than those farther away.

$$\mathbf{m}_i\,(t+1) = \frac{\sum_{j=1}^{n} h_{bmu_j,i}(t) \cdot \mathbf{x}_j}{\sum_{j=1}^{n} h_{bmu_j,i}(t)} \tag{4}$$

**Using the Batch Map for the User Interface** The batch map algorithm is stable with respect to repeated calculations performed on the same data set. This is why it has been chosen for the developed user interface. Since usability was one of the most important requirements, the user should not be compelled to learn totally new positions for the same pieces of music every time when a few songs are added to the repository.

### Aligned Self-Organizing Maps

Defining similarity is often a quite difficult task, which may involve several aspects. For example, images are distinguishable according to the used colors, shapes, textures or other criteria. These aspects can be extracted from different low-level features in various ways. Furthermore, they can be weighted differently and also compared on the basis of diverse metrics.

Generating one SOM for each of the different aspects raises the problem that the resulting SOMs are difficult to compare directly since the same data items are located in different regions of the map and also the cluster structure differs heavily. Addressing this issue, *Aligned SOMs* as introduced in [Pam03, PDW03] offer the possibility of gradually shifting the focus from one aspect to another by providing a number of aligned views. More precisely, multiple SOMs are trained on the same data using slightly but gradually modified parameters. The resulting stack consists of the SOMs that represent the two extreme values of the aspects and a number of SOM layers that are inserted between them to allow smooth transitions since neighboring SOM layers project same data items to similar regions.

Like the standard SOM, also the Aligned SOMs can be trained either sequentially or using batch training. However, to align the SOMs during training, it is necessary to define a distance between layers that determines the smoothness of the transitions between them. Given this distance, it is possible to calculate the pairwise distances between arbitrary items within the complete stack. The inter-layer distances, i.e. the distances between units of different layers, are used to align the layers in the same way the intra-layer distances between units within a map are used to preserve the topology of the data space.

The sequential training process is basically the same as that for the standard SOM. In the first step of each iteration, a data item $\mathbf{x}$ and a layer $l$ are randomly selected. Hereafter, the BMU for $\mathbf{x}$ is calculated within the chosen layer. The adaptations of the model vectors within layer $l$ are calculated based on the intra-layer distances exactly as shown in Expression 3. The update function for all other layers takes into account the inter-layer distances and adapts the model vectors according to the representation of the data item in the respective layer. As for the representation of the same data item in different layers, each data item is assigned one feature vector $\mathbf{x}_l$ for each layer $l$, where each $\mathbf{x}_l$ is composed of at least two feature sets (one for each aspect), which are weighted differently according to the feature balance of the layer. After having updated all model vectors in all layers, the described process is repeated iteratively until a defined convergence or some other stop criterion.

As for the batch training version of the Aligned SOMs, a very good explanation can be found in [PDW03]. However, since the calculation of aligned maps requires considering the relations between a large number of map units and different representations of same data items, the Aligned SOMs are computationally quite complex.

**A Simpler Approach of Aligned Maps for the User Interface** Using the developed *Matlab*®-program in order to create a hierarchical user interface for a given music repository involves calculating a large number of visualizations on different hierarchy levels. Therefore, a simpler and less time-consuming approach was chosen to generate multiple SOMs for differently weighted feature sets. This approach involves a new form of codebook initialization. In particular, given an already calculated SOM, its neighboring (with respect to the feature balance) SOMs are initialized by taking the model vectors of the existing one as their codebook. Although this is a very simple approach, it usually yields smooth transitions between neighboring SOMs (cf. Figure 6).

#### Smoothed Data Histogram (SDH)

The *Smoothed Data Histogram (SDH)* as proposed in [PRM02b] as a cluster visualization method for SOMs, aims at estimating and visualizing a probability density of the high-dimensional data items on the map. This estimation is based on a voting mechanism of the underlying multivariate feature vectors. Given a spread parameter $s$, each data item votes for the $s$ map units whose model vectors best resembles the feature vector of the data item. Taking into account the increasing distances to the $s$ BMUs, the closest map unit is assigned a value of $s$, the second closest a value of $s-1$, and so forth, until the $s$-th closest one is eventually assigned the value 1. All other map units receive 0 of this "similarity points". Moreover, the values of each rating are normalized by $\sum_{i=1}^{s} i$ in order to ensure that the sum of the votes equals 1 for each data item. s

After having processed all data items, the resulting distribution of the votes exhibits high values for regions on the map where the respective model vectors are similar to a large number of feature vectors. Therefore, visualizing this distribution shows typical clusters of the SOM. Since an important property of the SDHs are their smoothness, the distribution matrix is expanded by inserting interpolates between each pair of values in order to offer a more attractive view.

As for the influence of the spread parameter $s$ on the visualization, in the case of $s = 1$, the SDH equals the standard data histogram since only the BMUs are taken into account. With increasing values of $s$, the apparent clusters grow until they begin to merge and eventually result in only one big cluster at very high values of $s$.

## Prototype of the User Interface

The principal motivation for creating a user interface that visualizes the results of perceptual similarity measures was to support the user in exploring formerly unknown music. This would be difficult with traditional text-based search engines that are used by the majority of content providers of actual DMD-systems. Since users do not always know how to specify what they are seeking, nor even what they are looking for, developing solutions that address this issues is an important and challenging task [Pac03].

To generate the user interface for a given repository, a *Matlab*®-program, which processes the available data and finally creates a set of linked HTML-files and pictures was developed. HTML and JavaScript were used to ensure the independence from the operating system since web browsers supporting JavaScript are available for nearly all platforms.

The remainder of this section is organized as follows. In the first subsection, the data sources that can be used by the code generating *Matlab*®-program are reviewed. Hereafter, the structure and design of the user interface are presented as well as its functions. Subsection  then illustrates the different parts of the user interface that was generated based on the data of the test repository. Finally, the last subsection provides some results of a short usability study that was conducted to reveal shortcomings and gather suggestions for improvement.

## Available Data Sources

In this section, the various data sources that can be exploited to generate the user interface for a given repository are discussed.

#### Similarity Measures

The similarity measures form the most important data source since the SOMs are based on them. The RP/MFS-measure is chosen to calculate the rhythmic features for the user interface. The spectrum histograms (SH) model the timbral aspects.

#### User-Defined Directory Structure

Most users organize their music repositories with respect to some individual ontology. For this reason, they often create a directory structure that consists of folders for different genres, artists, albums or other criteria. This user-defined directory structure is taken into account by recursively accessing all directories of a given repository and creating visualizations for every visited folder. Regarding the SDH-visualizations of the user interface, for each piece of music that is not situated on a map which already represents the content of the directory containing the piece, a link to the appropriate folder enables the user to browse according to his/her familiar directory structure.

#### ID3-Tags

The ID3-tags of all MP3-files contained in the repository are extracted by a *Matlab*®-program which creates a file in every directory of the repository. This file consists of ID3-tags [6] of all music files that reside either in this folder or in directories at deeper levels.

#### Meta-Information from External Databases

A database containing categorizations for each piece of music in the test repository was created by the author in order to illustrate the visualization of arbitrary meta-information. In detail, every piece of music has been assigned a value for the following attributes:

- mood (values: sad, neutral, happy)
- tempo (values: very slow, slow, medium, fast, very fast, varying)
- complexity (values: low, medium, high)
- emotion (values: soft, neutral, aggressive)
- focus (values: instruments, vocals, both)
- genre, subgenre, subsubgenre (values taken mainly from descriptors of the *All Music Guide* [7])

The usage of external meta-data for creating the user interface is not restricted to a specific database format. A *Matlab*®-program is used to import the data from arbitrary external databases into *Matlab*®.

## Structure and Design

The structure and design of the user interface were elaborated in accordance with the most common principles for data visualization, namely *focusing* and *linking* [BMMS91]. These concepts and their application to the developed ViSMuC-interface are explained in the second part of this subsection. First, the functions and visualizations that are provided by the user interface are described.

[6] The following ID3-tags are used: *title*, *artist*, *album*, *year*, *genre*, *comment*, *bitrate*, *bitrate2*, *playtime*. In case of variable bitrate encoding the attributes *bitrate* and *bitrate2* indicate the minimum and maximum data throughput, respectively. If the encoding is performed using a constant bitrate, *bitrate* and *bitrate2* are the same.

[7] *http://www.allmusic.com*

**The Different Parts and Functions of the User Interface**

To get a first impression of the user interface, the reader is invited to take a look at Figure 1. This figure shows the three main parts of the ViSMuC-interface: control panel, main visualization area for SOMs, and meta-data visualization area.

The leftmost frame contains the control panel (cf. Figure 2) that is used to change the content of the other two areas. This control panel is split into four parts. At the very top of it, three navigation buttons can be found. Since the "back"- and "forward"-buttons of all popular Internet browsers are incapable of updating more than one frame at a single click, correctly working functions to go back and forward one view are provided by the leftmost and the rightmost of the three buttons, respectively. By clicking on the center button the user can always jump to a standard view of the root level directory that uses the colormap *"islands"* and is based solely on the rhythmic features. The metaphor of arrows as symbols on the navigation buttons was chosen because most users are familiar with it since it is very common. Below the navigation buttons, the feature balance selector is situated. Depending on the chosen number of aligned SOMs, the influence of either rhythmic and timbral features on the visualization can be adjusted gradually with this selector. Moving the mouse slowly from the topmost blue square over the intermediate links to the lowermost square results in loading aligned SOMs that successively shift their focus from rhythmic to timbral properties of the music. The next part of the control panel is the colormap selector, which is used to change the appearance of the actually displayed SOM by applying different color models.

Occupying the most space on the screen, the main visualization area, situated in the center frame, is used to display the SDH-visualizations. Furthermore, some important additional information can be found above the graphical representation: the root directory on which the actual visualization is based, the feature balance, and the current hierarchy level. As for the images of the SOMs/SDHs, displaying a grid on the map leads to easily distinguishable map units. The labels showing the song titles are truncated to either 15 or 20 characters depending on the total number of map units in order to fit into the grid elements. They directly link to the corresponding MP3-files. Furthermore, moving the mouse over a label opens a pop-up window containing the full name of the piece of music as well as additional information gained from its ID3-tags. An example of such an *"id3-tag info"*-box can be found in Figure 5. As for the red and yellow squares on the map, their function is explained in the next subsection.

The rightmost frame of the user interface represents the meta-data visualization area. Here, the distribution of attribute values given, for example, by ID3-tags or external databases is illustrated. This is accomplished by counting the number of songs that satisfy a certain (attribute, value)-assignment for each map unit and visualizing a smoothed version of the resulting quantity matrix. These meta-data visualizations support identifying the clusters formed by the SOM.

**Using Focusing and Linking in the Hierarchical Structure**

The concept of creating easy to understand illustrations each of which focuses on a particular aspect of the underlying data is usually referred to as focusing [BMMS91].

Very common focusing techniques are *selecting subsets* and *dimensionality reduction*. Both are applied each time a ViSMuC-user interface is created. Dimensionality reduction is achieved by using the data projection techniques PCA and SOM, whereas subset selection mainly aims at choosing those pieces of music that are displayed on each SOM. In the developed *Matlab*®-program, the cardinality of such a subset is determined by two factors: the number of map units of the SOM and the number of songs mapped to each unit.

Since the number of map units should be dependent on the number of data items, the map size is determined by taking the square root of the cardinality of the data set and multiplying it with a constant value. The result is rounded to obtain a column/row-ratio of 3:2. Moreover, there is a minimum map size of $2\times3$ since creating smaller maps does not make sense and furthermore would violate the constraint given by the column/row-ratio. Also the maximum number of map units is limited by a constant of the program that forces greater maps to reduce their size to either 54 or 96 map units, which leads to $6\times9$- or $8\times12$-SOMs, respectively. This was necessary in order to avoid visual overloading of the user as a result of displaying too many song titles on a single map. The presented approach for determining the map size works very well for the investigated repositories that contained between 15 and 834 pieces of music. Due to the size restrictions it is also appropriate for larger collections.

As for the number of songs that are projected to each map unit, it has been decided to display a maximum of five on a single unit. Nevertheless, the user can identify the real quantity by considering the number at the lower left corner of each map unit. If more than five pieces of music are projected to a certain map unit, the best matching data item in the respective Voronoi set, i.e. the song with the minimum Euclidean distance between its feature vector and the model vector, is chosen to represent a prototype of the map unit. Since this selection usually hides great parts of the repository, the omitted pieces of music have to be made available to

the user by other views. For this purpose, each Voronoi set containing more than five pieces is visualized by a new SOM that is situated on a lower hierarchy level. The need for connecting the different hierarchy levels accounts for the second design principle – linking.

In general, a consequence of focusing is that each view only presents partial information about the underlying data. Connecting these single views by inserting links between them is crucial to form a coherent image of the whole data. In the ViSMuC-interface, views of different hierarchy levels are linked by either yellow and red squares at the bottom of the map units at the higher level SOMs. While the red links point to those SOMs that were generated because the number of songs represented by a single map unit exceeded five, the yellow ones offer a convenient way to browse the directory in which the displayed song is stored. Hence, the red links connect the SOMs according to the hierarchy formed by musical properties, the yellow ones according to the directory structure.

In Figures 3 and 4 the results of the focusing and linking techniques, as described above, can be seen.

## Visualization of the Test Repository

As for the test repository, it consists of 834 MP3-files in 81 directories. The total play length is 3 666 minutes, thus about 61 hours. The music covers a wide range of different genres and styles.

Basically, all images produced by the ViSMuC-program are stored in the *Portable Network Graphics (PNG)* format [RPea99] since it combines lossless compression with small file sizes, even for truecolor images. Furthermore, using a color depth of 24 bits was crucial to preserve smooth color shadings.

### Aligned SOMs/SDHs

An example of aligned SOMs that are visualized by SDHs can be found in Figure 6, which reveals the changing cluster structure when the feature balance is shifted gradually from 100 percent rhythm to 100 percent timbre.

Comparing the two extreme views, the different clustering criteria become obvious. While the RP/MFS-measure clusters the pieces of music according to reoccurring activations in each of the 20 frequency bands, the SH-measure takes into account the intensity and recurrency of sounds that are quantized according to the critical-bands. Therefore, the SOM that is based solely on the results of the RP/MFS-measure projects songs with similar rhythm patterns, e.g. frequently reoccurring strong beats at low frequencies, to similar locations on the map. In contrast, the map which was generated exclusively on the basis of the timbral SH-features arranges the pieces of music according to the similarity of their spectral shapes.

Taking a closer look at the rhythmic perspective, it can be observed that the clustering coincides quite well with a distinction by genre.

Analyzing the SOM based on the SH-measure reveals one huge island which occupies the center and right regions of the map. Another much smaller one is spread along the leftmost two columns. Basically, these two islands differ in regard to the emotions the respective pieces of music are likely to invoke. In fact, while moving from the right areas of the map to the left ones, an increase in aggressiveness (and also in loudness) is noticeable. The peninsula with the little mountain which resides in the lower right corner is composed mainly of very soft songs – e.g. tracks from the albums *"Kuschelrock Vol. 11"*, *"Celtic Myths"* and *"Mystera IX"* – while the leftmost island on the map primarily represents Techno and Trance music – e.g. *"Thunderdome IV"* or *"Frankfurt Beat Productions"*.

### Colormaps

To address the varying preferences of different users in regard to the visual representation of the music repository, more precisely the visualization of the SDHs, three very dissimilar colormaps are made available.

**Islands** The colormap denoted as *"islands"* is a modified version of the one used in [Pam01]. It has been slightly adapted in order to better resemble the usual color scale of printed maps. However, the idea of emphasizing the transitions between seas and islands by inserting a "beach level" was preserved.

Basically, areas with few pieces of music mapped to them form oceans and lakes on the map, thus being colored in shades of blue. The darker the blue, the fewer songs are represented by the area. Those songs which lie in such regions are mostly outliers and often differ heavily from the main clusters which are illustrated by islands. As already mentioned, the borders between water and land are colored yellow since they represent beaches. For the clusters themselves the color scale covers a range from dark green (dense woods) to light green (light forests and veldts) to brown (hills) and finally to hues of gray and white (glaciers and snow-covered mountain tops).

**Fire** This newly created colormap emphasizes regions with many votes according to the calculation of the SDH (cf. Subsection ). Dark colors ranging from black to red are used in nearly two thirds of the available shades in order to suppress areas with few votes. The remaining third is a color gradient from orange to yellow. Due to the glowing appearance of the maps visualized with this colormap, it was named *"fire"*.

**Jet** Providing the highest contrasts between neighboring color levels, the colormap *"jet"* is capable of visualizing even small differences in the probability distribution calculated by the SDH. It is a standard colormap of the *Matlab*®-environment whose colors begin with dark blue, range through shades of blue, cyan, green, yellow and red, and end with dark red.

#### Distribution of Meta-Data Values

Visualizing meta-data – for example, those gained from ID3-tags or external databases – is accomplished by using an approach that is commonly known as *component planes* [KNK98]. A component plane visualizes the influence of each variable in the feature set on the cluster structure of the SOM. Since (attribute, value)-pairs can be considered as features, it is possible to apply the same technique, which has already been explained in Subsection .

In Figure 7 an assortment of component planes is presented. The leftmost group shows the distribution of the genres *"Classical"*, *"Rock"* and *"Electronica"* according to the ID3-tags. It can be observed that classical music is mapped exclusively to the island with the high mountain, which is situated in the upper right corner of the map. Furthermore, the lowermost component plane reveals that electronic music can be found in regions either at the lower right and the lower left. The other four groups of component planes show the distributions of some attribute values from the manual categorization. The illustrated attributes are *emotion*, *tempo*, *complexity* and *focus*. Analyzing these distributions, several interesting correlations between some of the attribute values can be observed. For example, the ID3-genre *"Classical"* coincides with neutral emotion, slow tempo, low complexity, and strong vocal appearance. Another interdependence can be stated between electronic music and focus on instruments, although the instruments used in this kind of music are mostly virtual.

## Usability Considerations

Since usability is a key-feature, especially for commercial DMD-systems, a small qualitative usability study surveying three persons has been conducted in order to reveal possible shortcomings. Its setup and results are presented in this subsection.

According to [GB99], information exploration activities can be characterized by the three dimensions of *users*, *tasks/goals* and *environment*. Each dimension is assigned a value that varies from *"real"* to *"synthetic"*. All possible combinations of values for each of these dimensions form the design space for evaluation experiments.

#### Users

As for the participating persons, neither of them is a music expert but all enjoy listening to music of various genres. Two of the test persons can be regarded as computer experts since they are advanced in their studies of computer science, whereas the third one has just basic knowledge in this field. Furthermore, each of the test persons stated that a system for exploring music collections by using different graphical visualizations would be useful. Therefore, they can be considered as real users.

#### Tasks/Goals

The following tasks and goals were elaborated.

1. Find music of the genre *"electronica"*.
2. Find soft pieces of music as well as aggressive ones.
3. Find all songs by the artist *"Nightwish"* and also some similar pieces of music.
4. Try out the different colormaps. Which one do you prefer?
5. Investigate different settings for the feature balance. Can you observe remarkable changes when the focus is shifted from rhythm to timbre?

The first three tasks illustrate typical queries a user may want to raise when searching for music. Hence, these tasks could be regarded as real. The fourth issue on the list takes into account the personal taste of the test persons. Eventually, the fifth one aims at examining the usefulness of presenting different views according to musical properties.

**Environment**

The evaluation was carried out using the complete test repository composed of 834 pieces. Since a large number of songs contained therein were completely unknown to the test persons, the setting is quite real according to the dimension *environment* – assuming the user interface is used for commercial DMD-systems with hundreds of thousands of different pieces of music. The ViSMuC-interface for the usability study was generated utilizing all available meta-data visualizations and three different views with respect to the feature balance.

**Results**

Tasks 1 and 2 were completed quickly and successfully by all test persons making intensive use of the meta-data visualizations to interpret the map. However, it was very interesting to observe the different approaches of the experienced computer users and the novice. While the former used the trial-and-error method, the actions performed by the latter were more intended and planned. In fact, the experienced users discovered the functions of the system by clicking on all that seemed to be a link. In contrast, the novice was a bit afraid of doing something wrong. After an introduction to the system, however, the novice performed the mentioned tasks efficiently without unnecessary clicks.

Task 3 – finding songs by *"Nightwish"* – turned out to be more difficult. Since neither of the three test persons knew any songs by *"Nightwish"*, they had moved the mouse over a lot of labels to view the ID3-tags before they finally succeeded. A possible solution to this problem would be to display another set of component planes that illustrates the distribution of the songs according to their artists. As for issue 4, while both of the computer experts were in favor of the colormap *"islands"*, the third test person preferred *"jet"* due to its high color contrasts. Finally, the results of the last task are quite disappointing since the different feature balances rather confused the test persons than supported them in gaining new insights.

# References


[BMMS91] A. Buja, J. A. McDonald, J. Michalak, and W. Stuetzle. Interactive Data Visualization using Focusing and Linking. In *Proceedings of the IEEE Visualization '91 Conference*, pages 156–163, 1991.

[GB99] G. Golovchinsky and N. J. Belkin. Innovation and Evaluation of Information Exploration Interfaces: A CHI98 Workshop. In *Association for Computing Machinery (ACM) Special Interest Group: Computer-Human Interaction (SIGCHI) Bulletin*, volume 31, pages 22–25, January 1999.

[Hot33] H. Hotelling. Analysis of a Complex of Statistical Variables Into Principal Components. *Journal of Educational Psychology*, 24:417–441 and 498–520, 1933.

[Jol86] I. T. Jolliffe. *Principal Component Analysis*. Springer, New York, 1986.

[KNK98] S. Kaski, J. Nikkilä, and T. Kohonen. Methods for Interpreting a Self-Organized Map in Data Analysis. In M. Verleysen, editor, *Proceedings of ESANN'98, 6th European Symposium on Artificial Neural Networks*, pages 185–190. D-Facto, Brussels, Belgium, April 1998.

[Koh82] T. Kohonen. Self-Organizing Formation of Topologically Correct Feature Maps. *Biological Cybernetics*, 43:59–69, 1982.

[Koh92] T. Kohonen. New developments of learning vector quantization and the self-organizing map. In *Symposium on Neural Networks; Alliances and Perspectives in Senri (SYNAPSE 1992)*, Osaka, Japan, 1992. Senri International Information Institute.

[Koh01] T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, 3rd edition, 2001.

[Pac03] F. Pachet. Content Management for Electronic Music Distribution: The Real Issues. *Communications of the Association for Computing Machinery (CACM)*, April 2003. To Appear.

[Pam01] E. Pampalk. Islands of Music: Analysis, Organization, and Visualization of Music Archives. Master's thesis, Vienna University of Technology, Austria, December 2001.

[Pam03] E. Pampalk. Aligned Self-Organizing Maps. In *Proceedings of the Workshop on Self-Organizing Maps*, Kitakyushu, Japan, September 2003.

[PDW03] E. Pampalk, S. Dixon, and G. Widmer. Exploring Music Collections by Browsing Different Views. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR'03)*, Washington, D.C., USA, October 2003.

[PRM02a] E. Pampalk, A. Rauber, and D. Merkl. Content-based Organization and Visualization of Music Archives. In *Proceedings of the Association for Computing Machinery (ACM) Multimedia*, pages 570–579, Juan les Pins, France, December 2002. ACM.

[PRM02b] E. Pampalk, A. Rauber, and D. Merkl. Using Smoothed Data Histograms for Cluster Visualization in Self-Organizing Maps. In *Proceedings of the International Conference on Artifical Neural Networks (ICANN'02)*, pages 871–876, Madrid, Spain, August 2002. Springer.

[RPea99] G. Randers-Pehrson and et. al. PNG (Portable Network Graphics) Specification, Version 1.2. *http://www.libpng.org/pub/png/spec/1.2/index.html*, July 1999.

[SJ02] U. Seiffert and L. C. Jain. *Self-Organizing Neural Networks, Recent Advances and Applications*. Physica-Verlag Heidelberg, 2002.

[Ves00] J. Vesanto. Using SOM in Data Mining. Licentiate's thesis at the Helsinki University of Technology, Finland, April 2000.


Figure 1: The user interface for the root directory of the test repository (hierarchy level 0) incorporating a SOM with 54 map units. The left frame represents a control panel, the centered one exhibits the actual SDH-visualization, and that at the right displays information about the distribution of meta-data values over the map.

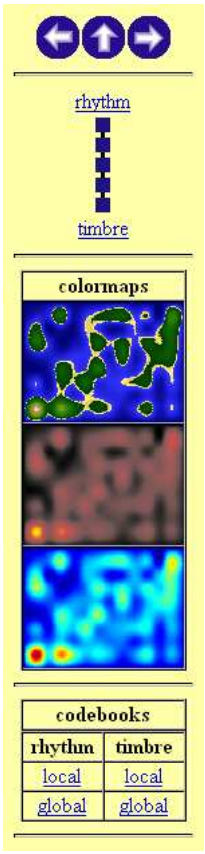


Figure 2: A close view of the complete control panel. From top to bottom: navigation buttons, feature balance adjustment, colormap selector, links to codebook visualizations.


Figure 3: A close view of 6 map units. The number in the lower left corner of each unit indicates the quantity of songs represented by it. If this number is greater than 4, a map containing only the pieces of the particular unit can be accessed by clicking on the red square. The yellow squares are links to maps of those directories where the displayed tracks reside.

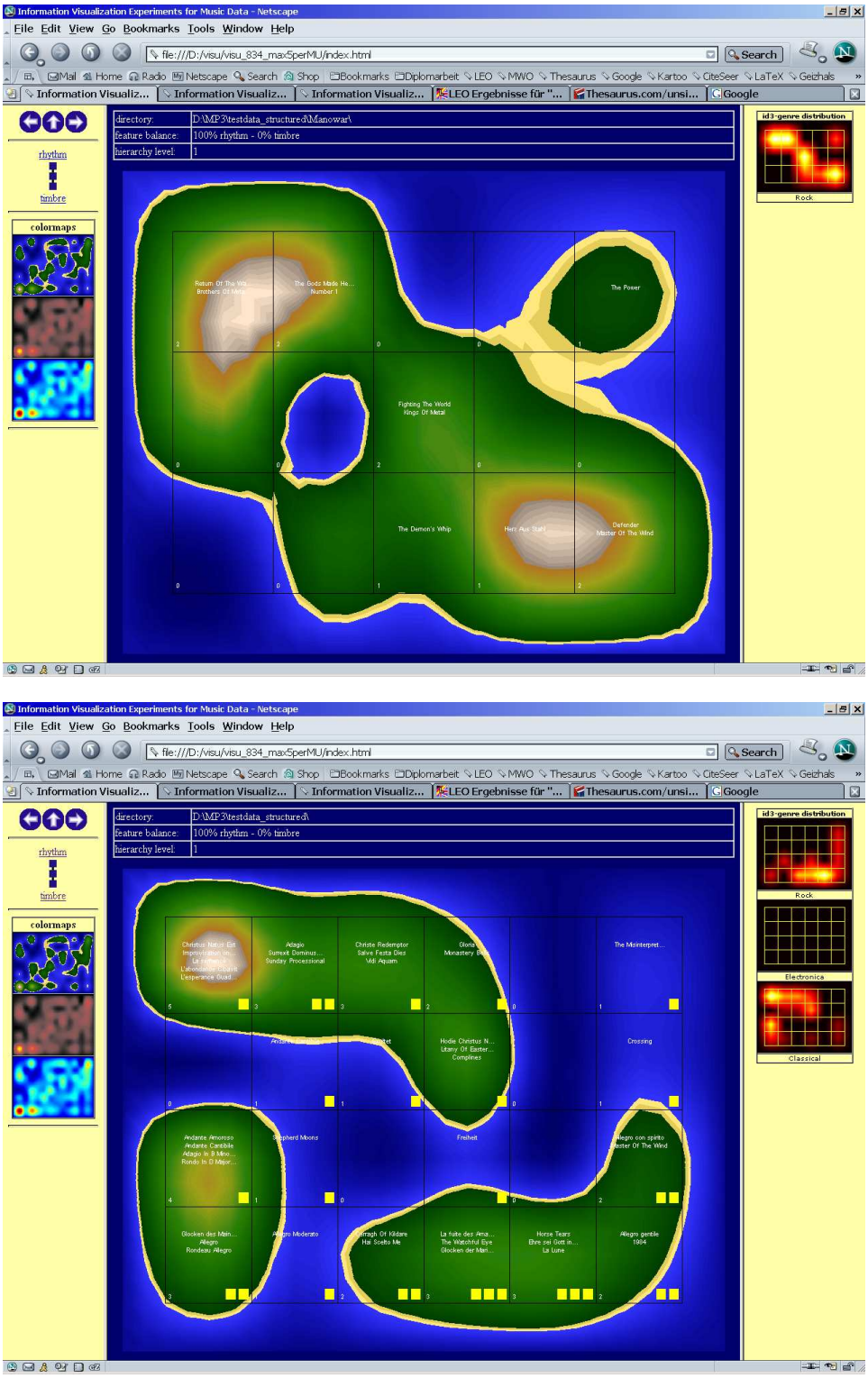


Figure 4: Depiction of two SDHs in hierarchy level 1, which are both accessible through links of the map unit in hierarchy level 0 (cf. Figure 1) whose prototype is *"Master of the Wind"* (situated at the very lower left). The upper visualization was created according to the directory structure of the repository, thus showing the contents of the folder *"Manowar"*, where the mentioned prototype song resides. The lower one contains a view showing all pieces of music that are projected to the same map unit as the prototype. Hence, this view is based on the results of the similarity measures.

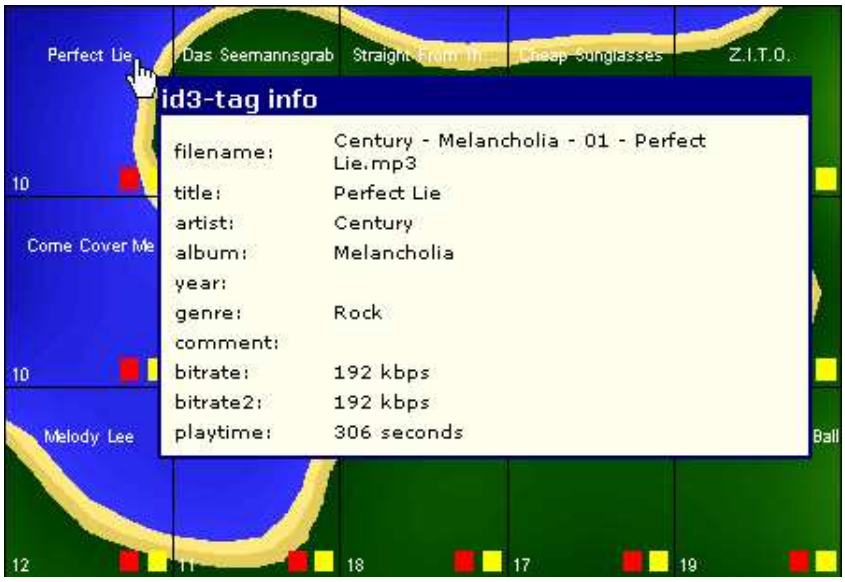


Figure 5: Example of the pop-up window that appears when the user moves the mouse over the label of an arbitrary piece of music. In this case, the ID3-information of the respective song is displayed.

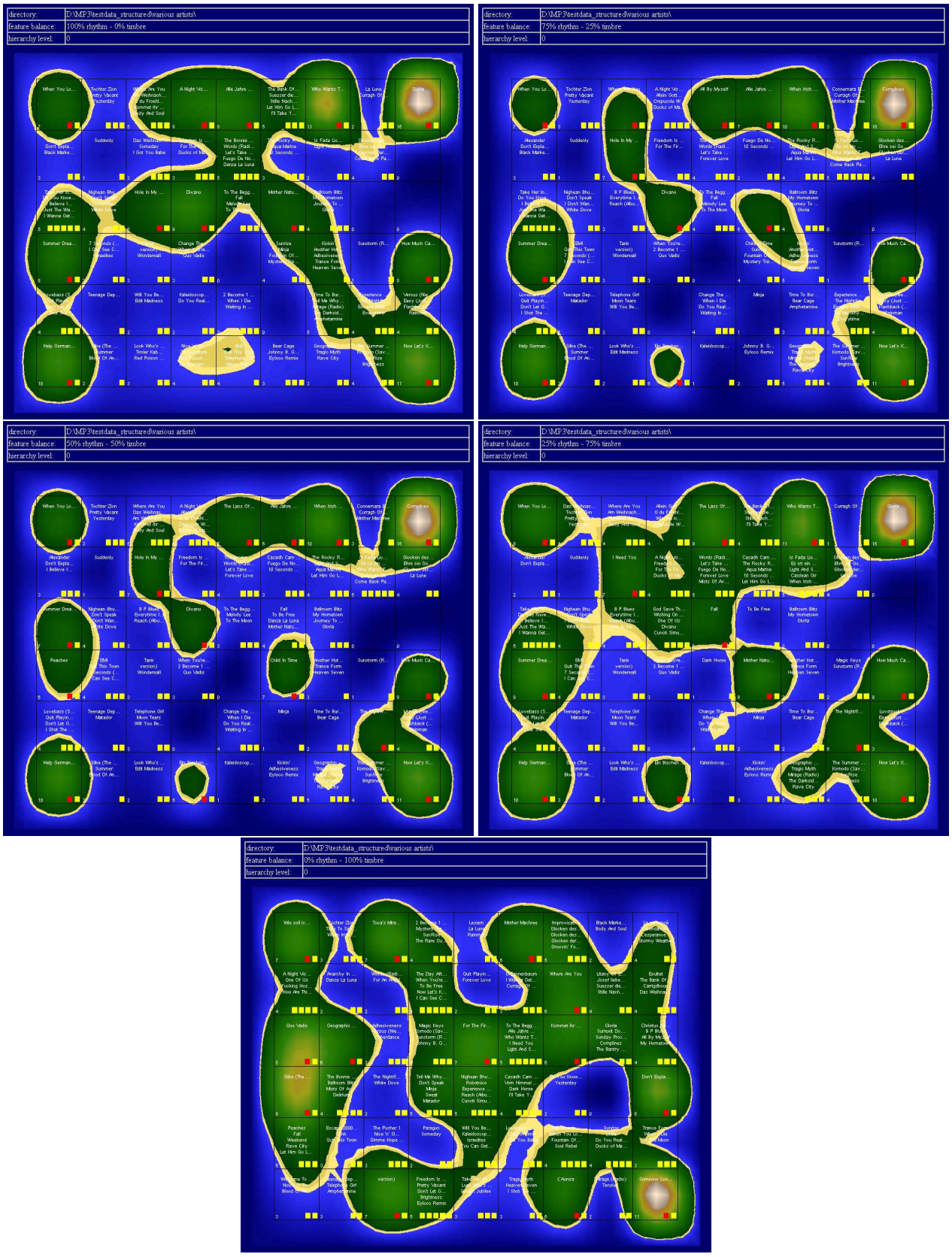


Figure 6: Illustration of the modifications of the cluster structure when the focus is gradually shifted from 100 percent rhythm to 100 percent timbre in 5 steps (100/0, 75/25, 50/50, 25/75, 0/100). The user interface was created using a 6×9-SOM and taking *"Various Artists"* as root directory.

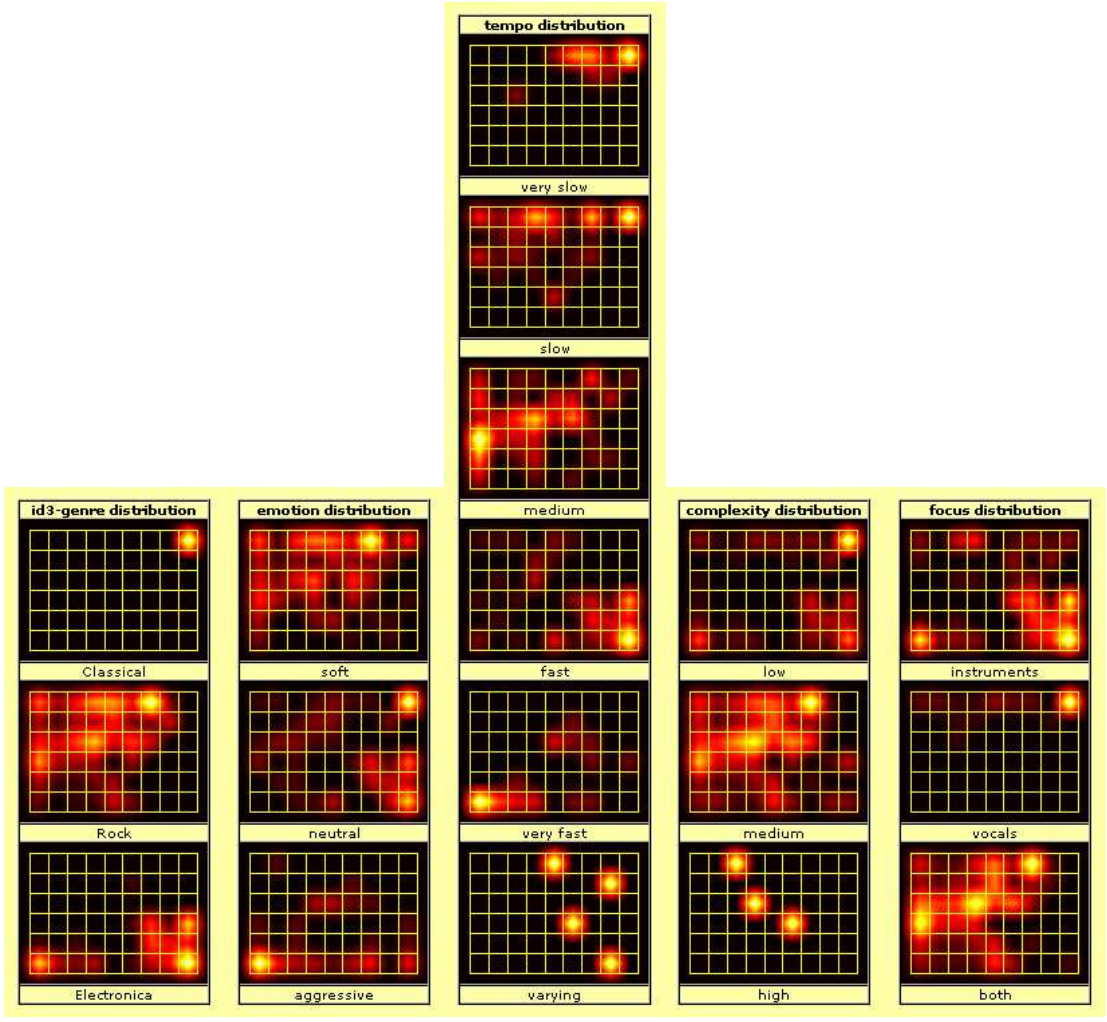


Figure 7: Illustration of distributions of some attribute values. The leftmost picture visualizes the distribution of the values assigned to the ID3-tag *genre*. The other images provide information about some of the attributes that were used in the manual classification.