Technical Report: The CoMIRVA Toolkit for Visualizing Music-Related Data

Markus Schedl*
Department of Computational Perception
Johannes Kepler University Linz, Austria

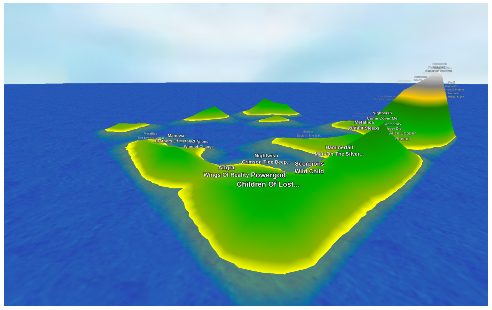

Figure 1: Our *muscape* application which uses the CoMIRVA framework.

Abstract

We present *CoMIRVA*, which is an abbreviation for *Collection of Music Information Retrieval and Visualization Applications*. CoMIRVA is a Java framework and toolkit for information retrieval and visualization that is licensed under the GNU General Public License. At the moment, the main functionalities of the toolkit include music information retrieval, web retrieval, and visualization of the extracted information.

In this paper, we focus on the visualization aspects of CoMIRVA. Since many of the information retrieval functions are intended to be applied to problems of the field of music information retrieval (MIR), we demonstrate the functions using data like similarity matrices of music artists gained by analyzing artist-related web pages. Currently, CoMIRVA supports the following visualization techniques: Self-Organizing Map grid, Smoothed Data Histogram, Circled Bars, Circled Fans, Continuous Similarity Ring, and Sunburst. We present the corresponding functions with a special focus on the Sunburst visualization of term occurrence matrices, which we have implemented lately.

Since one key feature of CoMIRVA is its easy extensibility, we further elaborate on how we used it for creating a novel user interface to digital music repositories.

1 Introduction and Context

Music information retrieval (MIR) has become an important field of research over the past few years. Its main concern is the extraction, analysis, and representation of information that describes various aspects of music. This information can be gained basically from three kinds of sources:

- audio signal of digital music files
- metadata provided by music distributors
- information extracted from the web

While descriptive high-level features based on signal analysis usually describe properties like rhythmical structure (e.g. [16, 7]) or timbral aspects (e.g. [2, 13]) of a piece of music, features based on metadata incorporate, for example, ID3-tags. Information extracted from artists' web pages or gathered by collaborative filtering, in contrast, not only relies on expert opinions like metadata provided by music distributors, but reflects a kind of cultural or community knowledge because it incorporates the opinions of a large number of people (e.g. [26, 9, 10, 19]).

Important aspects of MIR include the usage and representation of the extracted information. Usually, the features are used to derive similarities between music artists or pieces of music, which is one of the most important tasks in MIR and is supported by CoMIRVA with a wide variety of functions.

Possible applications that are based on MIR include recommender systems or metadata-enriched music players that may present song lyrics or images related to the artist currently played. Since building such applications from scratch requires extensive knowledge and is very time-consuming, we have developed (and are constantly extending[1]) CoMIRVA in order to facilitate this process. The aim is to provide a fully functional system for Music Information Retrieval and Visualization, which can be used as a toolkit[2] and as a framework for building specialized applications[3]. To achieve this twofold purpose, CoMIRVA is based on an object-oriented design concept and implemented in Java, which also facilitates its extensibility.

This paper is the first publication to present CoMIRVA. We have, however, already published a very brief overview of the main visualization functions that had been implemented by then, at *IEEE Visualization 2005* in [22].

1.1 Related Systems

There exist several systems for performing MIR and InfoVis tasks independently. However, all of them have more or less serious drawbacks when it comes to combining tasks related to both MIR and InfoVis.

*e-mail: markus.schedl@jku.at

[1] This is accomplished by the author as well as by students in the context of practical courses and theses.

[2] For this purpose, the user is provided a GUI.

[3] We will elaborate on such an application in Section 5.

1.1.1 All-in-one Solutions

Professional all-in-one solutions like *Matlab*® can be applied for rapid prototyping (of MIR as well as of InfoVis applications), but suffer from the drawback of a very high price and an even higher complexity. However, there exists a large number of specialized toolboxes related to MIR and InfoVis. As an alternative to *Matlab*®, a few systems have been developed under the GNU General Public License[4], for example, *Octave*[5]. However, with respect to visualization functionality and usability, they cannot compete with *Matlab*®.

1.1.2 MIR Toolkits

In regard to their feature extraction functionality, MIR toolkits usually only provide audio-based signal extraction. One of the most popular MIR toolkits is *Marsyas*[6] (cf. [25]), which is a framework for rapid prototyping and experimentation and provides functions for retrieval, analysis, and synthesis of audio signals. *jAudio* (cf. [15]), as part of the *ACE*[7] project, is another signal-based feature extraction tool. A third popular framework that focus on research and application development in the audio and music domain is *CLAM*[8].

1.1.3 InfoVis Toolkits

Among the toolkits related to information visualization, the most popular ones include *Pad++*[9] (cf. [5]), *Piccolo*[10] (cf. [4]), formerly known as *Jazz* (cf. [6]), and the *InfoVis Toolkit*[11] (cf. [8]). Pad++ is mainly a toolkit for creating zoomable user interfaces and has been used, for example, to develop InfoVis applications like zoomable web browsers or image browsers. Piccolo is a quite popular 2D graphics framework for developing graphical applications in Java and C#. The InfoVis Toolkit is a relatively recent development which not only provides a set of different visualizations, but also special data structures that are well-suited for a number of information visualization tasks.

1.1.4 Motivation for CoMIRVA

The main motivation for starting to develop CoMIRVA was the high price and very inefficient memory management of *Matlab*® which we used before for music information retrieval and visualization tasks. Furthermore, we wanted to structure a number of algorithms that we had already implemented in Java and embed them in a larger framework.

CoMIRVA is novel in two regards. First, we are not aware of any solution that combines MIR and visualization techniques within one framework. Indeed, CoMIRVA's visualization functions are suited to fulfill the special requirements of tasks related to MIR, for example, special data structures like similarity matrices. Second, to the best of our knowledge, we do not know any MIR system that includes web mining functionalities.

1.2 Outlook

The remainder of this paper is organized as follows. Section 2 gives a general overview of CoMIRVA and presents its functionalities related to data I/O and data manipulation. The following two sections, 3 and 4, elaborate on the music information retrieval and visualization functions, respectively. To demonstrate its easy usage as an InfoVis and MIR framework, we present an application that heavily relies on CoMIRVA's functions in Section 5. Finally, Section 6 summarizes the work and gives a short outlook on possible future extensions.

2 General Overview of CoMIRVA

CoMIRVA was designed using the object-oriented design paradigm and was implemented in Java, which facilitates its extensibility. Since CoMIRVA is open source and thus everybody who is interested may extend it, this was an important requirement. The reason why we have chosen Java as programming language is its platform independence and its good support for network access which is obviously crucial for web mining.

As CoMIRVA is not only a framework of MIR and InfoVis algorithms, but also intends to serve as a toolkit, it offers a GUI through which most of the functions provided by the framework are accessible.

2.1 Data Types and Data I/O

CoMIRVA basically provides two data types: *data matrices* and *meta-data vectors*. Data matrices are arbitrarily sized $m \times n$ matrices of double precision values, meta-data vectors are ordered lists of strings that usually describe the rows or columns of a data matrix. Both data matrices and meta-data vectors can be assigned a name under which they are displayed in separate lists in the right part of the GUI (cf. Figure 2). Naturally, the user can also rename each data item. As a matter of course, CoMIRVA supports loading and saving of data matrices and meta-data vectors from/to standard text files. Moreover, CoMIRVA also provides saving and loading of *workspaces*, i.e. collections of data matrices and meta-data vectors. This facilitates easy restoring of associated data.

2.2 Data Manipulation

As for data manipulation, the current version of CoMIRVA provides *normalization* of data matrices as well as a function which we call *vectorization*. Normalization can be performed linearly or logarithmically to a range whose boundaries are defined by the user (by default to [0,1]). Furthermore, the user can set the scope of the normalization. This determines whether the minimum and maximum values of the complete matrix are mapped to those given by the normalization boundaries, or the minima and maxima are determined for every row or every column separately and therefore also mapped independently row- or column-wise to the defined boundaries. Since we often work with matrices that indicate similarities or distances between each pair of a number of entities, e.g. music artists, normalization in the scope of (independent) similarity vectors (matrix by row or column) is useful. Another task we often have to perform is the decomposition of a similarity matrix into its single similarity vectors. For this purpose, a function which we call vectorization is provided by CoMIRVA. This vectorization can be performed by row or by column and creates as many new 1-dimensional data matrices as rows or columns are present in the original data matrix. As a further convenient feature of the vectorization, each newly created data vector is named after the original data matrix and the description of the respective row or column as given by the meta-data vector (if one is selected). As a result, each similarity vector can easily be assigned a context.

[4] *http://www.gnu.org/copyleft/gpl.html*
[5] *http://www.octave.org/*
[6] *http://opihi.cs.uvic.ca/marsyas/*
[7] *http://coltrane.music.mcgill.ca/ACE/*
[8] *http://www.iua.upf.es/mtg/clam/*
[9] *http://www.cs.umd.edu/hcil/pad++/*
[10] *http://www.cs.umd.edu/hcil/piccolo/*
[11] *http://ivtk.sourceforge.net/*

3 MUSIC INFORMATION RETRIEVAL

As already mentioned, information about music artists or pieces of music can be gained from the audio signal or from metadata provided by the music distributor or extracted from the web. Since web mining techniques have successfully been applied to MIR problems like similarity measurement (e.g. [26]), artist-to-genre classification (e.g. [9, 19]) or automatic lyrics detection ([10]) and we actively participate in this field of research, we have implemented some simple web mining functionalities. Furthermore, CoMIRVA offers state of the art functions for feature extraction based on the audio signal. The provided functionalities for both types of information retrieval approaches are presented in the following.

3.1 Co-occurrence Analysis

Performing co-occurrence analysis of artist names on web pages, i.e. determining which artists are mentioned together on the same web page, was proposed in [27] for finding related artists to a given seed artist and in [19] for deriving complete similarity matrices which were used successfully for artist-to-genre classification. Given a list of artist names (or arbitrary other entities), CoMIRVA provides a function that uses an arbitrary search engine to estimate the number of web pages containing each artist and each pair of artists. These page counts are inserted in a (symmetric) page count matrix, which is added to CoMIRVA's GUI after the process has finished. Based on such a page count matrix, CoMIRVA can calculate a conditional probability matrix that estimates the probability for the name of an artist (or another entity) A to occur on a web page which is known to mention another artist (or entity) B. This probability matrix represents a complete similarity matrix which can be used for a wide variety of applications in MIR such as prototypical artist detection (cf. [20, 21]) or artist-to-genre classification (e.g. [19]). A more detailed elaboration of the topic of co-occurrence analysis for MIR can be found in [19].

3.2 Term Profile Creation

The simple co-occurrence analysis as described above does not take the content of web pages into account as it relies only on the page counts provided by the search engine. However, it is often desirable to analyze the content of web documents. To this end, CoMIRVA offers functions for automatic retrieval of web documents (HTML files), extraction of terms from these (or from other text documents), and calculation of some measures used in text retrieval. We have implemented a special data structure called *Entity Term Profile (ETP)* that uses XML to describe the content of a single or a set of documents. More precisely, such an ETP contains a list of terms that were automatically extracted from the document(s) as well as the paths to the document(s), which are necessary for using ETPs in interfaces for document search, like our Sunburst implementation. In the case of an ETP describing a set of documents instead of a single one, term occurrences, term frequencies, document frequencies, and the well-established $TF \times IDF$ (term frequency $\times$ inverse document frequency) values (cf. [18]) are stored additionally.
To summarize, the functions provided by CoMIRVA for creating and processing ETPs comprise:

- retrieving web documents using arbitrary search engines
- creating a term list comprising all terms that occur in the retrieved documents
- generating an ETP from the retrieved documents and storing (serializing) it in an XML file
- loading XML-serialized ETP files into CoMIRVA's GUI
- creating a Sunburst-like user interface for document search based on an ETP, cf. Section 4.3.1

3.3 Audio-based Features

Features derived from the audio signal of a piece of music range from very simple low-level properties like zero crossing rate, spectral centroid, or spectral flux to sophisticated high-level descriptors that model the rhythmical or timbral structure of a piece. We have integrated some of the most successful high-level features in CoMIRVA.
The rhythm-based *Fluctuation Patterns* were first presented in [16]. They model the periodicity of the audio signal for a number of critical frequency bands (according to the bark scale) and periodicity intervals (in beats per minute) and also incorporate a model of human auditory perception. The outcome of a Fluctuation Pattern calculation on a piece of music is a feature vector whose dimensionality depends on the number of bark intervals and periodicity intervals. To use a set of such feature vectors for defining similarities between pieces of music, e.g. the Euclidean distances between the feature vectors must be calculated.
Furthermore, two different feature extraction algorithms that are based on *Mel Frequency Cepstral Coefficients (MFCCs)* are implemented in CoMIRVA. MFCCs give a coarse description of the envelope of the frequency spectrum and thus, model timbral properties of a piece of music. Since MFCCs are calculated on time invariant frames of the audio signal, usually, *Gaussian Mixture Models (GMMs)* are used to model the MFCC distributions of a whole piece of music. Similarity between two pieces of music A and B is then derived by drawing a sample from A's GMM and estimating the probability that this sample was created by B's GMM.
CoMIRVA offers two MFCC-based similarity measures. The first corresponds to the one described in [3] (called *Aucouturier and Pachet* in CoMIRVA), the second corresponds to [14] (called *Mandel and Ellis*). The measures basically differ in terms of the number and type of GMMs used and in calculation time.
Given a directory, CoMIRVA recursively searches for MP3 files and calculates the requested audio features for all of them.

4 INFORMATION VISUALIZATION

The implemented functions for information visualization can be categorized according to the type of input data they use. We differentiate between algorithms that work on feature data, those working on similarity matrices or similarity vectors, and those working on special data structures, like term occurrence matrices in the case of our Sunburst visualization of ETPs.
Each visualization provided by CoMIRVA is implemented in its own class, but has to be connected to an instance of the class *VisuPane* which is responsible for double buffering and serves as interface between the individual visualizations and CoMIRVA's GUI. The visualization classes also implement a mouse listener if user interaction is desired, e.g. in the Circled Fans or Sunburst interface. To each visualization, a *colormap*, i.e. a mapping from a range of values to a range of colors, can be applied. However, the influence of the chosen colormap on the visualization varies according to the visualization type. Furthermore, visualizations can be saved as PNG or JGP files for later use.

4.1 Visualizations of Feature Data

This kind of data usually represents high-dimensional high-level descriptors. In the case of MIR data, these might be rhythmic or timbral properties of music, e.g. the periodicity of the audio signal for a number of frequency bands (in Hertz) and periodicity intervals (in beats per minute), cf. Section 3.3.

4.1.1 Self-Organizing Map (SOM)

The *Self-Organizing Map (SOM)*, e.g. [11, 12], is a well-established unsupervised neural network that aims at clustering high-dimensional data items in a usually 2- or 3-dimensional space such that similar data items are mapped to similar regions of the target space. CoMIRVA currently supports four different initialization methods: Random, Gradient, Linear (cf. [12]), and SLC (cf. [24]). Furthermore, the size of the SOM grid and the training length can be adjusted by the user.[12] Sequential (online) training is supported as well as batch training.
As for the visualization of a SOM grid, after a SOM has been trained, each data item is mapped to the map unit that best represents it. This unit is called the *best matching unit (BMU)*. Determining the BMU for every data item and drawing the SOM grid and the names of the data items on their respective BMU yields visualizations like the one in Figure 2 (without the colorful cluster visualization). This figure shows a SOM trained on web features of music artists. The upper left regions of the SOM contain mainly artists that create quite aggressive music. In the lower right, a peninsula with electronic music can be found. The other artists are mostly mapped to the big islands in the lower left.

4.1.2 Smoothed Data Histogram (SDH)

A visualization approach that emphasizes the data clusters of a SOM is the *Smoothed Data Histogram (SDH)*, proposed in [17]. An SDH estimates the density of the data items over the map. To this end, each data item votes for a fixed number of best matching map units. The selected units are weighted according to the quality of the matching. The votes are accumulated in a matrix describing the distribution over the complete map. After each piece of music has voted, the resulting matrix is interpolated in order to obtain a smooth visualization. Finally, the interpolated matrix is visualized by applying a colormap. An example of an SDH visualization can be found in Figure 2, where the colormap *Islands* was applied to give the impression that clusters of similar artists form islands which rise from the blue sea (the sparse areas of the SOM).

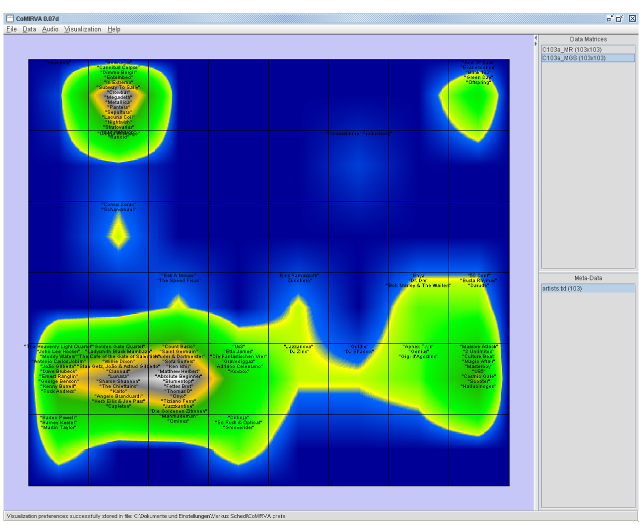

Figure 2: A Smoothed Data Histogram (SDH) visualization of a Self-Organizing Map (SOM) trained on features of music artists.

[12]By default, a simple heuristic is used for determining a suitable size for the SOM.

4.2 Visualizations of Similarity Vectors and Matrices

Similarity vectors describe how similar a number of items (e.g. music artists) are to a given one. Similarity matrices indicate the similarity between all pairs of items of a given item set. In the following, we present some visualizations provided by CoMIRVA that help the user to find music artists or pieces of music which are similar to a given one.

4.2.1 Circled Bars

The *Circled Bars* visualization approach offers a simple method to answer questions like: "Which artists produce similar music to that of my favorite artist *A*?". It thus takes a similarity vector as input. Given a seed artist *A*, an adjustable number of most similar artists (according to the used similarity measure) are arranged in a circle. The artists are ordered by their similarity to artist *A*. The similarity values are visualized by filled arcs that vary in length and color corresponding to the applied colormap. Figure 3 shows a sample visualization with artists similar to the Metal band *Stratovarius*. For this figure, the Circled Bars visualization was generated from co-occurrences (cf. Section 3.1), and the colormap *Fire* was applied. Hence, the values in parentheses after the artist names indicate the probability for the respective artist to be found on a web page that is known to mention the seed artist *A*. Since the Circled Bars visualization does not require high computing or graphics capabilities, it may serve as a user interface for small devices with limited screen size, like mobile phones or personal digital assistants.

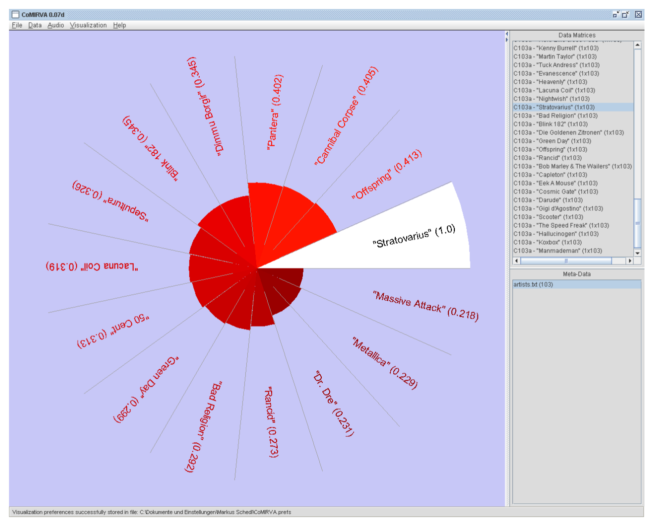

Figure 3: A Circled Bars visualization of a vector describing similarities between music artists.

4.2.2 Circled Fans

The *Circled Fans* visualization is a conceptual extension of the simple Circled Bars. While the Circled Bars only take the nearest neighbors of a given seed artist (or any other entity) into account, the Circled Fans incorporate similarities in a transitive manner.
Given a seed artist *A* whose name is displayed in the center of the visualization, an adjustable number of most similar artists are arranged in a circle around *A* and connected to *A* by edges whose thickness and color correspond to the similarities given by the similarity matrix and the chosen colormap. The thicker the connecting edge, the more similar two artists are. Subsequently, for each of the similar artists of *A*, again, the most similar ones are selected, arranged in a circular arc whose center is the respective parent node,

and connected to this parent node by an edge.
The user can adjust the maximum edge thickness, the maximum number of data items on the inner circle and on the outer circular arcs (which we call fans), as well as the angular extent of the fans. Moreover, the Circled Fans support user interaction by redrawing the visualization with a new seed artist B whenever the user clicks on the label of an arbitrary artist B.
In Figure 4, a screenshot of a Circled Fans visualization with the seed artist *Evanescence* is depicted. In this example, an asymmetric similarity matrix derived from co-occurrences (cf. Section 3.1) was used to define artist similarity, and the colormap *Colorful* was applied. Visualizing such asymmetric similarities is an important area of application of the Circled Fans. For example, the Circled Fans depicted in Figure 4 reveal that the band *Green Day* is mentioned on 53% of the web pages containing the word "Evanescence", whereas "Evanescence" can only be found on 21% of the web pages that mention "Green Day". Such information about similarity asymmetries can be used for measuring the popularity of an artist and further for determining which artists are prototypical for a specific genre (cf. [20]).

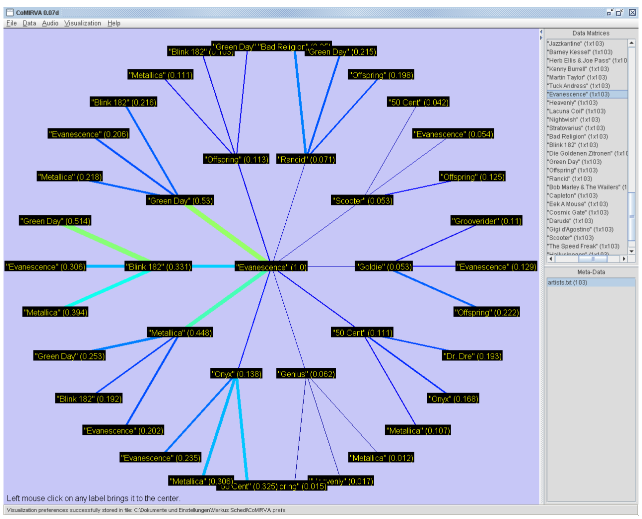

Figure 4: A Circled Fans user interface of a matrix describing similarities between music artists.

4.2.3 Continuous Similarity Ring (CSR)

The *Continuous Similarity Ring (CSR)* visualization technique, which is described in detail in [20], uses a graph-based model for illustrating similarities between entities (e.g. music artists) by using one prototype for each of a number of given classes (e.g. music genres). Since prototypical artists are usually very well known, they can serve as reference points for finding similar, but less known artists, e.g. in online music stores.
Given a set of artists and information on which artist belongs to which genre, we determine a prototype for each genre and arrange these prototypes in a circle, cf. Figure 5. Since similar or related prototypes and the genres they represent should be placed close to each other, we formulate a *Traveling Salesman Problem* on the distance matrix generated from the artists' similarity matrix and apply a simple heuristic algorithm. The resulting tour defines the arrangement of the artists within the circle of prototypes.
Additionally, for each prototypical artist, an adjustable number of its most similar neighbors (according to the used similarity matrix) are shown. To preserve the distances given by the similarity matrix, the neighbors are positioned using a cost-minimizing heuristic. The artists' vertices are connected by edges whose thickness and color vary according to their similarity values and the colormap applied. For the visualization depicted in Figure 5, the colormap *Fire* was applied.

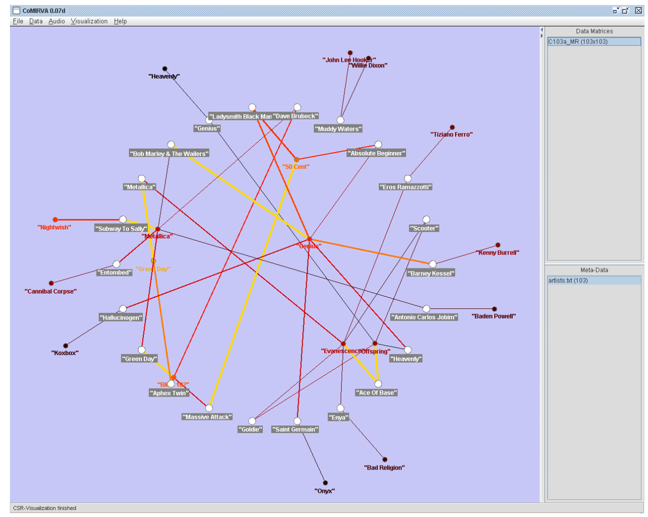

Figure 5: A Continuous Similarity Ring (CSR) for visualizing prototypical entities. Here, the entities are music artists and the prototypicality of each artist is calculated for his/her assigned genre.

4.3 Visualizations of Term Occurrence Matrices

As already mentioned in Section 3.2, CoMIRVA provides a data type called Entity Term Profile which describes a set of documents by various properties relevant for information retrieval tasks. Among other data, an ETP contains a term occurrence matrix that indicates in which documents every term of a given term list occurs.

4.3.1 Sunburst

We use such term occurrence matrices to create a user interface that makes use of the well-established *Sunburst* visualization technique (cf. [1, 23]).
Starting with the whole set of documents, more precisely, the ETP that describes this set, the terms with the highest document frequency are selected and visualized as filled arcs around a centered circle (the root node) that represents the entire document collection. The size (angular extent) of each individual arc is proportional to the document frequency of the associated term, i.e. to the number of documents containing the term. Performing the term selection with respect to document frequencies recursively for all arcs eventually yields a complete Sunburst visualization. Internally, CoMIRVA stores the Sunburst as a tree. Every arc A represents a set of documents that contain the term associated with A and the terms associated with all arcs that must be traversed on the shortest way to the root node.
The user can define a number of stop criteria to limit the calculation time, the size of the Sunburst, and the number of recursions. CoMIRVA provides the following constraints: maximum sub nodes per node, maximum depth of the tree, minimum angular extent of an individual arc. The font size of the labels, i.e. the terms, is automatically adapted to the angular extent of the arc. However, minimum and maximum values for the font size can be defined by the user.
Since our Sunburst interface is intended to be used for document search, user interaction is provided in two ways. First, clicking with

the left mouse button on an arbitrary arc generates a new Sunburst visualization with this arc as root node, i.e. only the documents that are represented by the selected arc are used. Second, a right mouse click on any arc displays a pop-up menu with the locations of the documents represented by the selected arc. The user can then view a document by selecting it from the pop-up menu.
Figure 6 shows a screenshot of a Sunburst interface generated from an ETP of web documents about the music artist *Louis Armstrong*. The values in parentheses indicate the document frequency. This sample visualization reveals which terms occur in a collection of web documents about *Louis Armstrong*. If the user wants to know, for example, in which documents *Louis Armstrong* and *Miles Davis* are mentioned together, s/he can easily display a list of these documents by clicking on the respective arc as shown in Figure 6. A further click on one of the documents opens it in the standard web browser.

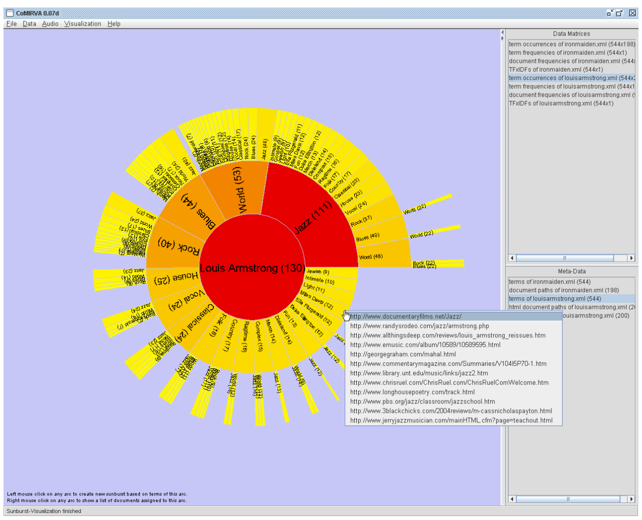

Figure 6: A user interface for finding (web) documents, which is based on the Sunburst visualization technique. The data sources of the visualization are term occurrence matrices that are generated from a set of documents.

5 An Example Application based on CoMIRVA

Since the beginning of the development of CoMIRVA in late 2004, a number of students have participated in extending the framework. For example, we elaborated a class/interface structure that facilitates extending the audio-based feature extraction algorithms in collaboration with one of our master students, who further implemented two well performing feature extractors (cf. [16, 3]).
Easy extensibility and easy usage in other applications were vital requirements when designing the class structure and interfaces of CoMIRVA. One application that we have developed recently and that heavily makes use of functionality provided by CoMIRVA is *muscape*[13], which provides a novel, innovative user interface to music repositories.
Given an arbitrary collection of digital music files, *muscape* automatically extracts features from the audio signal and trains a SOM on them to form clusters of similar sounding pieces of music. Subsequently, the distribution of the pieces of music on the SOM is determined using an SDH. We interpret this SDH as a three-dimensional height profile and visualize it as a landscape applying a colormap that resembles that of geographical maps. This geographical metaphor, which is called *Islands of Music* (cf. [16]), yields a landscape where sparse areas are represented by oceans (in blue) whereas clusters with many pieces of music look like mountains (brown and gray) that rise from islands (green).
Since similar pieces of music are mapped to similar regions on the landscape by the SOM, the user can intuitively explore his/her own or someone else's music collection by moving through the landscape like in a 3D game. The angle of the viewport is automatically adjusted according to the height of the current position, i.e. if the current position in the landscape is directly in front of a high mountain, the user has the feeling that s/he glances at the top of the mountain; if s/he, in contrast, resides on top of the mountain, the view is adjusted to see which songs are situated at the mountain's foot.
While the user explores the terrain, s/he is presented an anisotrop auralization of the music. More precisely, the four songs that are nearest to the current position in the landscape are played simultaneously with their volumes adjusted to the distances, and the direction from which the sound seems to come adjusted to the direction in which the respective song resides on the landscape.[14]
In Figure 7, some screenshots of a landscape generated from music from the genres Electronic and Metal are depicted. While the images in the first row show the appearance of the landscape and the automatic viewport adaptation, those in the second row illustrate special modes in which, instead of the artist and track names, descriptive terms and images are presented. These terms and images are automatically extracted from the web using only the artist names as input.

6 Conclusions and Future Work

In this paper, we presented CoMIRVA, an open-source framework and toolkit for music information retrieval and visualization. It combines functions for feature extraction (directly from the audio signal of digital music files as well as from metadata that is derived from the world wide web), special data structures like similarity matrices and Entity Term Profiles, and information visualization approaches that are suited to visualize music-related data. CoMIRVA is implemented in Java and freely available under *http://www.cp.jku.at/comirva*. We further elaborated on our application called *muscape* that demonstrates CoMIRVA's suitability as a framework for building own applications.
As for future work, there are many directions into which CoMIRVA should be extended. For example, hierarchical visualization techniques to deal with arbitrarily sized music collections are desirable. Also, time series visualization approaches that describe changes of properties of a piece of music over time should be integrated. Moreover, simple tools like a colormap editor or a statistical editor for data matrices would further increase CoMIRVA's usability. We continuously keep extending CoMIRVA with the help of students that are interested in music information retrieval and information visualization.

7 Acknowledgments

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

[13] *muscape* mainly relies on the following functions provided by CoMIRVA: calculation of SOM and SDH, audio feature extraction, various functions for web mining.

[14] This anisotrop auralization requires a 5.1 or 7.1 surround system.

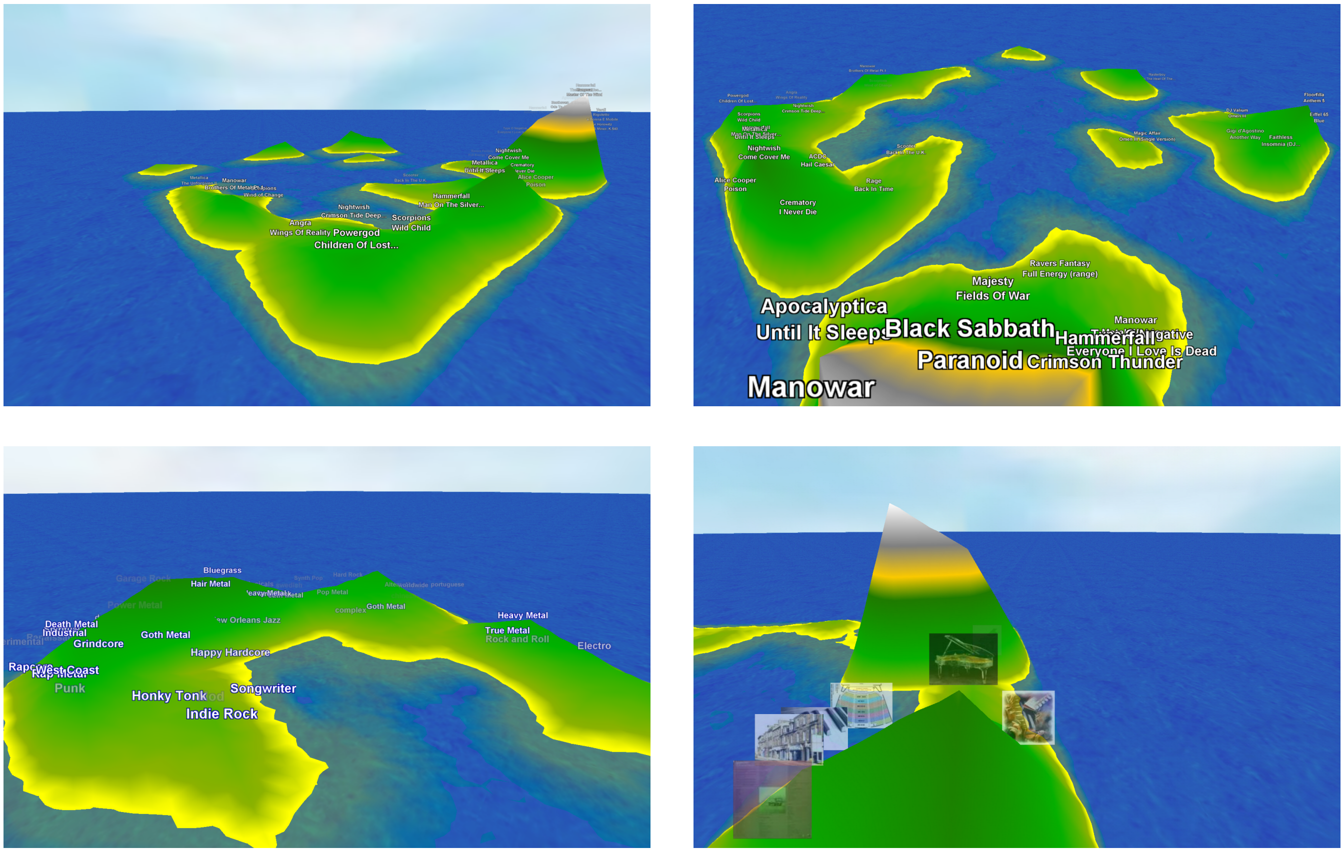

Figure 7: Some screenshots of our *muscape* application. The upper left image shows a global view of a landscape generated from Metal and Electronic music. The upper right image illustrates the automatic adaptation of the viewport angle when positioning on a mountain. On the lower left image, an island showing terms that describe the respective artists instead of their names is depicted. Finally, the lower right image shows a landscape enriched with images that describe the music in the particular regions. Both the terms and the images are automatically extracted from the web given only the artist names.

CoMIRVA, especially Klaus Seyerlehner, who implemented high-level feature extractors and Markus Straub, who participated in the development of initialization methods for Self-Organizing Maps.

References

[1] Keith Andrews and Helmut Heidegger. Information Slices: Visualising and Exploring Large Hierarchies using Cascading, Semi-Circular Discs. In *Proceedings of IEEE Information Visualization 1998*, pages 9–12, Research Triangle Park, NC, USA, October 1998.

[2] Jean-Julien Aucouturier and François Pachet. Music Similarity Measures: What's the Use? In *Proceedings of the 3rd International Symposium on Music Information Retrieval (ISMIR'02)*, Paris, France, October 2002.

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

[4] Benjamin B. Bederson, Jesse Grosjean, and Jon Meyer. Toolkit Design for Interactive Structured Graphics. *IEEE Transactions on Software Engineering*, 30(8):535–546, August 2004.

[5] Benjamin B. Bederson, James D. Hollan, Ken Perlin, Jonathan Meyer, David Bacon, and George W. Furnas. Pad++: A Zoomable Graphical Sketchpad For Exploring Alternate Interface Physics. *Journal of Visual Languages and Computing*, 7(1):3–32, 1996.

[6] Benjamin B. Bederson, Jon Meyer, and Lance Good. Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology (UIST'00)*, pages 171–180, San Diego, CA, USA, November 2000.

[7] Simon Dixon, Elias Pampalk, and Gerhard Widmer. Classification of Dance Music by Periodicity Patterns. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR'03)*, pages 159–166, Baltimore, MD, 2003. John Hopkins University.

[8] Jean-Daniel Fekete. The InfoVis Toolkit . In *Proceedings of the 10th IEEE Symposium on Information Visualization (InfoVis'04)*, Austin, TX, USA, October 2004.

[9] Peter Knees, Elias Pampalk, and Gerhard Widmer. Artist Classification with Web-based Data. In *Proceedings of the 5th International Symposium on Music Information Retrieval (ISMIR'04)*, pages 517–524, Barcelona, Spain, October 2004.

[10] Peter Knees, Markus Schedl, and Gerhard Widmer. Multiple Lyrics Alignment: Automatic Retrieval of Song Lyrics. In *Proceedings of 6th International Conference on Music Information Retrieval (ISMIR'05)*, pages 564–569, London, UK, September 2005.

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

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

[13] Beth Logan. Mel Frequency Cepstral Coefficients for Music Model-

ing. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR'00)*, Plymouth, Massachusetts, USA, 2000.

[14] Michael I. Mandel and Dan P. W. Ellis. Song-Level Features and Support Vector Machines for Music Classification . In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, London, UK, September 2005.

[15] Daniel McEnnis, Cory McKay Ichiro Fujinaga, and Philippe Depalle. jAudio: A Feature Extraction Library . In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, London, UK, September 2005.

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

[17] Elias Pampalk, Andreas Rauber, and Dieter 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.

[18] Gerard Salton and Christopher Buckley. Term-weighting Approaches in Automatic Text Retrieval . *Information Processing and Management*, 24(5):513–523, 1988.

[19] Markus Schedl, Peter Knees, and Gerhard Widmer. A Web-Based Approach to Assessing Artist Similarity using Co-Occurrences. In *Proceedings of the 4th International Workshop on Content-Based Multimedia Indexing (CBMI'05)*, Riga, Latvia, June 2005.

[20] Markus Schedl, Peter Knees, and Gerhard Widmer. Discovering and Visualizing Prototypical Artists by Web-based Co-Occurrence Analysis. In *Proceedings of the Sixth International Conference on Music Information Retrieval (ISMIR'05)*, London, UK, September 2005.

[21] Markus Schedl, Peter Knees, and Gerhard Widmer. Improving Prototypical Artist Detection by Penalizing Exorbitant Popularity. In *Proceedings of the Third International Symposium on Computer Music Modeling and Retrieval (CMMR'05)*, Pisa, Italy, September 2005.

[22] Markus Schedl, Peter Knees, and Gerhard Widmer. Interactive Poster: Using CoMIRVA for Visualizing Similarities Between Music Artists. In *Proceedings of the 16th IEEE Visualization 2005 Conference (Vis'05)*, Minneapolis, Minnesota, USA, October 2005.

[23] John Stasko and Eugene Zhang. Focus+Context Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualizations. In *Proceedings of IEEE Information Visualization 2000*, pages 57–65, Salt Lake City, UT, USA, October 2000.

[24] Mu-Chun Su, Ta-Kang Liu, and Hsiao-Te Chang. Improving the Self-Organizing Feature Map Algorithm Using an Efficient Initialization Scheme . *Tamkang Journal of Science and Engineering*, 5(1):35–48, 2002.

[25] George Tzanetakis and Perry Cook. MARSYAS: A Framework for Audio Analysis. *Organized Sound*, 4(3), 2000.

[26] Brian Whitman and Steve Lawrence. Inferring Descriptions and Similarity for Music from Community Metadata. In *Proceedings of the 2002 International Computer Music Conference*, pages 591–598, Goeteborg, Sweden, September 2002.

[27] Mark Zadel and Ichiro Fujinaga. Web Services for Music Information Retrieval. In *Proceedings of the 5th International Symposium on Music Information Retrieval (ISMIR'04)*, Barcelona, Spain, October 2004.