CONTENT-BASED ORGANIZATION OF DIGITAL AUDIO COLLECTIONS

Robert Neumayer **Thomas Lidy** **Andreas Rauber**
Vienna University of Technology
Department of Software Technology and Interactive Systems
Favoritenstrasse 9-11/188, A-1040 Vienna, Austria
robert.neumayer@univie.ac.at, {lidy, rauber}@ifs.tuwien.ac.at

ABSTRACT

With increasing amounts of audio being stored and distributed electronically, intuitive and efficient access to large music collections is becoming crucial. To this end we are developing algorithms for audio feature extraction, allowing to compute acoustic similarity between pieces of music, as well as tools utilizing this information to support retrieval of as well as navigation in music repositories. This paper provides an overview of our research activities in the domain of music IR. It presents the Rhythm Patterns feature set and demonstrates its suitability for music genre recognition while also presenting an acoustic interpretation by re-synthesizing sound from that feature set. Furthermore, it outlines the principles of organizing digital music repositories using *Self-Organizing Maps* and presents the novel *PlaySOM* interface and the *PocketSOMPlayer* for mobile devices, both providing intuitively explorable music information spaces.

Keywords: content-based audio retrieval, audio feature extraction, music genre classification, Self-Organizing Map, music repository organization, clustering, navigation, user interfaces, mobile devices

1 INTRODUCTION

The increasing popularity and size of digital music repositories drives the need for advanced methods to organize those archives for both private as well as commercial use.

The ability to offer users information about songs or artists that are similar to the ones they were actually searching for, holds a great market potential as recommendation engines have proven. The linking of similar products to results of customer searches bears fruits, even when it is not based on the similarity of products themselves, but on the buying behavior of other customers, like Amazon.com has impressively shown. Therefore commercial music vendors could particularly profit from organization of music archives based on sound-similarity.

It is an intrinsic need for them to offer high-level user interfaces to their repositories to satisfy their customers' needs. Search methods based on track similarity, such as query-by-example, offer alternatives to keyword based searches that avoid the downside of having to rely on manually assigned metadata. A system that offers searches that rely on metadata only, can never meet these more sophisticated needs.

The heart of every audio-content-based music retrieval system is the audio feature extractor. Its purpose is to derive content descriptors from the audio signal that represent semantics of the musical content. Researchers in the domain of music information retrieval developed a range of different audio descriptors. Each of them may perform different depending on the specific retrieval task. In any case the result of the feature extractor is fundamental for tasks like similarity-based retrieval, automatic organization or classification of music.

In our work we focus on the Rhythm Patterns feature set and show its applications in genre classification tasks as well as automatic organization of large music archives using *Self-Organizing Maps* (*SOMs*).

A question frequently raised, particularly for non-standard feature sets, is on the cognitive characteristics of the extracted numbers. Thus, we present an acoustical re-synthesis of the feature set. With the audible feature set, one can evaluate the effectiveness of the feature extraction through asking a human for the same task as the computer, working only with the substantially reduced information from the aggregated descriptor, e.g.: Can you discriminate musical genres provided only with the information from the feature set?

The motivations for organizing private music collections are most likely fun and entertainment by overcoming the limitations of conventional media players. Similarity-based organization of music archives allows users to explore pieces of music that are similar to ones they know and like. Moreover, it provides a clear and easy navigation for music collections the users are familiar with and allows users to abstract from manually assigned genre information which is, at least in private collections, often inappropriate or simply missing.

Concerning the access to rapidly growing and changing collections, the similarity-based organization may prove much more satisfying than conventional search methods because users do not have to know new songs by name. This problem gets more important with the growing size of a collection. Browsing a few hundred songs a user knows well might not be much of a problem using metadata, but navigating through thousands of songs one is not familiar with may lead to restrictions, preventing the user from gaining access to the majority of songs.

We present two intuitive interfaces for accessing mu-

sic collections. The music tracks are organized spatially on a two-dimensional map display based on the similarity of the extracted sound features. We will show how this map metaphor is used to provide convenient access to music repositories and how such an organization of songs can be used for playlist generation and interactive exploration for both desktop applications and mobile devices.

The remainder of this paper is structured as follows. Section 2 briefly reviews related work. Section 3 explains the technical background of the audio content descriptors and mentions the process of re-synthesizing features to obtain an audible feature set. In Section 4 we report the performance of our approach in music genre classification. Section 5 gives a brief introduction to *Self-Organizing Maps* followed by a description of the results of automatic organization of a music collection. It furthermore describes the novel *PlaySOM* and *PocketSOMPlayer* user interfaces in detail. Finally, Section 6 provides some conclusions.

2 RELATED WORK

Content-based music analysis and its applications like similarity-based search and organization experienced a major boost in the late 1990's when mature techniques for the description of audio content became available. From that time on a range of researchers has been working on different methods for content-based audio retrieval. Almost as manifold as the feature computation approaches are their subsequent applications.

One of the first works to incorporate psycho-acoustic modeling into an audio feature extraction process and using the *SOM* for organizing audio data is reported in (Feiten and Günzel, 1994). A first approach to classify audio recordings into speech, music, and environmental sounds is presented in (Zhang and Zhong, 1995). Another work on classification of sounds into different categories is (Wold et al., 1996), applying loudness, pitch, brightness, bandwidth, and harmonicity features.

(Foote, 1997) presents a search engine which retrieves audio from a database based on similarity to a query sound. An early work on musical style recognition is (Dannenberg et al., 1997), which investigates various machine learning techniques applied for building style classifiers.

(Logan and Salomon, 2001) perform content-based audio retrieval based on K-Means clustering of MFCC features and define a novel distance measure for comparison of descriptors. The *MARSYAS* system (Tzanetakis and Cook, 2000, 2002) uses a wide range of musical surface features to organize music into different genre categories using a selection of classification algorithms.

(Pampalk et al., 2003) conduct a comparison of several content-based audio descriptors on both small and large audio databases, including a feature set called Fluctuation Patterns, similar to the Rhythm Patterns we use in our work.

A system performing trajectory matching using *SOMs* and MFCCs is presented in (Spevak and Favreau, 2002).

Regarding intelligent playlist generation, an exploratory study using an audio similarity measure to create a trajectory through a graph of music tracks is reported in (Logan, 2002). Furthermore, many applications can be found on the Internet that are not described in scientific literature. An implementation of a map-like playlist interface is the *Synapse Media Player*[1]. This player tracks the user's listening behavior and generates appropriate playlists based on previous listening sessions and additionally offers a map interface for manually arranging and linking pieces of music for an even more sophisticated playlist generation. Another example of players offering automatic playlist generation is the *Intelligent Multimedia Management System*[2] which is based on tracking of the user's listening habits and recommends personalized playlists based on listening behavior as well as acoustic properties like BPM or a song's frequency spectrum.

A novel interface particularly developed for small-screen devices, was presented in (Vignoli et al., 2004). This artist map interface clusters pieces of audio based on content features as well as metadata attributes using a a spring model algorithm. The need for advanced visualization to support selection of audio tracks in ever larger audio collectionis also addressed in (Torrens et al., 2004), where different representation techniques grouping audio by metadata attributes using Tree-Maps and a disc visualization are presented.

3 AUDIO FEATURE EXTRACTION

Audio Feature Extraction is at the core of every query-by-similarity, music organization or classification task. Its output is crucial for the subsequently applied methods. The goal of Audio Feature Extraction is to retrieve semantics from audio, that are able to characterize different styles of audio. While drastically reducing the amount of information from the plain audio wave data, feature extraction has to derive sufficient information suitable for describing the content of the audio. It is a great challenge to define what audio similarity really is. Even human's have great difficulties in agreeing upon various genre taxonomies or describing what makes two pieces of music similar. Thus, a general definition of sound similarity is not possible. Audio Feature Extraction intends to capture what a human listener hears when listening to music. Our approaches incorporate numerous psycho-acoustic transformation steps which are based on studies of the human auditory system. Still, the computer system is no equivalence to the human perception, nevertheless, the algorithms prove to be well suited for tasks such as automatic genre classification, similarity queries, organization of music archives, etc.

3.1 Rhythm Patterns

One of our main contributions to research in Music Information Retrieval is the Rhythm Patterns feature set, calculated from analysis of the spectral audio data, first introduced in (Rauber and Frühwirth, 2001), and later drastically enhanced by incorporating psycho-acoustic transformations (Rauber et al., 2002). The algorithm for extract-

[1] www.synapseai.com

[2] www.luminal.org

ing the Rhythm Patterns is a two stage process: First, from the spectral data, the specific loudness sensation according to the human auditory system is computed on various frequency regions, incorporating several psycho-acoustic phenomena. Second, the specific loudness sensation values are transformed into a time-invariant domain resulting in a representation of modulation amplitudes per modulation frequency (i.e. kind of energy or rhythm variation) on several frequency regions.

We will give an outline of all the steps involved in the feature extraction process. An overview is provided in Figure 1, a detailed description of a previous version of the algorithm was presented in (Rauber et al., 2003).

The algorithm processes audio tracks in standard digital PCM format with 44.1 kHz sampling frequency as input. Audio compressed with e.g. the MP3 format will be decoded in a preprocessing step. Each audio track is segmented into pieces of 6 seconds length. A short time Fast Fourier Transform (STFT) is applied to retrieve the energy per frequency band, i.e. the spectrum, every 11.5 ms, resulting in a spectrogram of the 6 second segment. To reduce the amount of data, the frequency bands of the spectrogram are summed up to 24 so-called critical bands, according to the Bark scale (Zwicker and Fastl, 1999). A further psycho-acoustical phenomenon incorporated is spectral masking, i.e. the occlusion of one sound by another sound. This phenomenon is coped with a spreading function. Successively, the data is transformed into the logarithmic decibel scale, equal-loudness curves are accounted for, resulting in a transformation into the unit Phon and afterwards into the unit Sone, reflecting the specific loudness sensation of the human auditory system. At this point, we retrieved the specific loudness sensation over time on 24 critical frequency bands. Still, we have a time-dependent signal, although reduced to 511 sample values at the time axis due to the window size in the STFT.

In order to obtain a time-independent representation of the data, another Fourier Transform is applied. The idea is to regard the varying energy on a frequency band of the spectrogram as a modulation of the amplitude over time. With the second Fourier Transform, the spectrum of this modulation signal is retrieved. It is a time-invariant signal that denotes the modulation frequency on the abscissa, and the magnitude of modulation on the ordinate. A high amplitude at the modulation frequency of 2 Hz for example indicates a strong rhythm at 120 bpm (beats per minute = modulation frequency * 60). The notion of rhythm ends above 15 Hz, where the sensation of roughness starts and goes up to 150 Hz, the limit where only three separately audible tones are perceivable. The algorithm captures modulation frequencies up to 43 Hz, however we cut off the information above a modulation frequency of 10 Hz. Subsequently, modulation amplitudes in that range are weighted according to a function of human sensation depending on modulation frequency, accentuating values around 4 Hz, followed by the application of a gradient filter and gaussian smoothing.

The final feature vector contains a time-invariant representation of fluctuation strength according to human sensation between 0.168 Hz and 10 Hz of modulation frequency on 24 critical frequency band regions. A feature

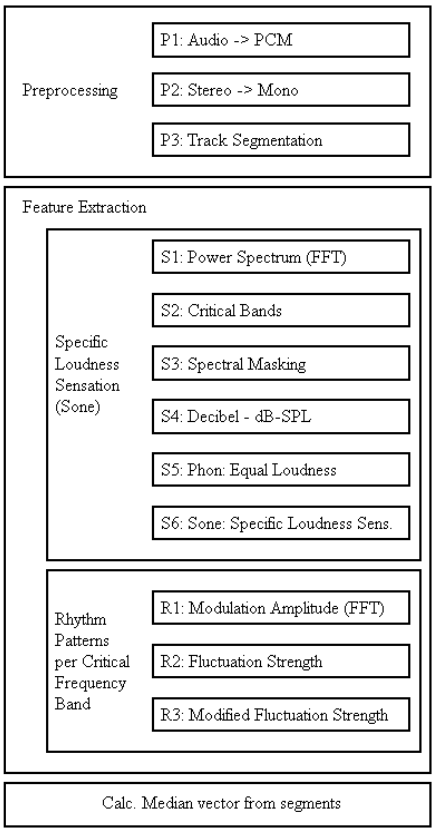


Figure 1: Audio feature extraction block diagram.

vector for each 6 second segment of a piece of music is calculated. In order to summarize the characteristics of an entire piece of music we simply average the feature vectors derived from its segments by computing the median. This approach extracts suitable characteristics of semantic structure for a given piece of music to be used for music similarity tasks.

3.2 Re-Synthesizing the Features

it is desirable to obtain a notion of the content of the features, in order to be able to assess the quality of the feature set and its applicability to a specific task. A way to give insight to the feature set is to make it audible. Apart from giving insight into the features, it provides a possibility for monitoring the feature extraction process, and additionally, it enables the monitoring person to check if he or she as a human would be able to discriminate between classes (such as genres) provided only with feature data.

We thus re-synthesize an audio signal from the *Rhythm Patterns* representation Lidy et al. (2005). An Inverse Fourier Transform is used for the synthesis of the modulation signal. As there is no exact information about the original signal, we take the centre frequency f_i of that critical band, i.e.

$$f_i = c_{i-1} + (c_i - c_{i-1})/2 \quad (1)$$

as the frequency of the base signal of critical band i, where c_i is the upper band limit of critical band i, c_0 being 0. The minimum modulation frequency is 0.168 Hz, so we will

have to re-synthesize a sound of 6 seconds length to accommodate 1 period of the lowest modulation frequency.

The output is the modulation signal of one frequency band, $m_i[t]$, $t \in N$. This signal can now be used to modulate the centre frequency of the critical band, f_i. We, however, do not know about the original amplitude of the signal on that band. We utilize the DC component of the modulation signal, which in general is >0, as the base amplitude A_i of the band signal. Therewith, we can modulate the band centre frequency f_i with the modulation signal. The same process is accomplished on all 24 critical frequency bands and the modulated signals are heterodyned, as in Eq. 2.

$$s[t] = \sum_{i=1}^{24} A_i \times cos(2\pi f_i t) \times m_i[t] \qquad (2)$$

The resulting signal $s[t]$ reflects the structure of the original piece of audio and resembles fluctuations within the critical bands as captured by the feature extraction process.

First evaluations showed, that the rhythmic structure of audio pieces can be recognized in the re-synthesized signal. Besides acoustically verifying the sound and comparing the rhythm of the re-synthesized sound with that of the original music through listening, the rhythmic structure can also be seen in the visualization of the waveform. Music where strong beats do not play an important role (e.g. classical music) can clearly be discriminated from other genres. When a specific rhythm in the form of drums and beats differs by definition from one genre to another genre (e.g. Hip-Hop versus Reggae versus Drum'n'Bass), the genres can easily be distinguished in the acoustical representation of the feature vectors.

3.3 Statistical Spectrum Descriptor

The Statistical Spectrum Descriptor (SSD) is computed during calculation of the Rhythm Patterns features. The spectrum which is available after the transformation into Bark scale and the appliance of several psycho-acoustic transformations represents rhythmic characteristics within the specific frequency range of the 24 critical bands. According to the occurence of beats or other rhythmic variation of energy on a specific band, statistical measures are able to describe the audio content. We intend to describe the rhythmic content of a piece of audio by computing the following statistical moments on the values of each of the 24 critical bands: mean, median, variance, skewness, kurtosis, min- and max-value. The Statistical Spectrum Descriptor (SSD) represents a smaller descriptor for rhythmic audio content and can be used as a single feature vector or in combination with other descriptors in subsequent tasks.

3.4 Rhythm Histogram Features

The Rhythm Histogram features are a descriptor for general rhythmics in a piece of audio. Contrary to the Rhythm Patterns and the Statistical Spectrum Descriptor, information is not stored per critical band. Rather, the magnitudes of each modulation frequency bin of all 24 critical bands are summed up, to form a histogram of "rhythmic energy" per modulation frequency. The histogram contains 60 bins which reflect modulation frequency between 0.168 and 10 Hz. For a given piece of audio, the Rhythm Histogram feature set is calculated by taking the median of the histograms of every 6 second segment processed.

4 MUSIC CLASSIFICATION

Audio descriptors derived in the Feature Extraction process build the basis for a range of different retrieval tasks. Besides being applicable directly in similarity based searches, audio descriptors are often deployed in artificial intelligence approaches. In that domain, the Music Information Retrieval approaches make use of both supervised and unsupervised machine learning techniques. While unsupervised learning approaches are valuable in automatic organization of music archives (see Section 5), supervised machine learning techniques are applied for automatic classification tasks. From a number of examples the computer learns how to classify music pieces into a number of previously defined classes. The taxonomy can be defined according to specific task requirements. In our work we performed classification into musical genres.

The advantage of music classification through supervised machine learning is, that - provided that annotated ground-truth data exists - the result of the learning process can be directly measured in terms of accuracy, precision and recall percentage values. Direct evaluation is not possible in unsupervised learning tasks, where the result of automatic organization is of subjective nature.

In this section we present evaluation results of our algorithms' performance in music genre classification task. In Section 4.1 we explain the methods we used for music classification. Section 4.2 describes the music collections involved in the genre classification experiments. Section 4.3 presents the results and improvements we made during experiments with music genre classification.

4.1 Classification Method

We utilize the Weka Machine Learning Software[3] as the environment for our genre classification task. The output from our Feature Extractor is converted to the Weka data format. As the machine learning algorithm we chose Support Vector Machines with pairwise classification. In order to get an assessment of the generalization of the approach we use a 10-fold cross validation for each experiment: The music collection is divided into 10 subsets, in each of the 10 iterations a different subset is chosen for testing and the other 90 % of the data is used for the training process. The cross validation result is the average of the 10 runs.

4.2 Music Collections

We conduct our experiments on three diferent music collections, which gives us an indication about the applicability of the approach do various different music repositories and thus different musical styles. The first audio

[3] http://www.cs.waikato.ac.nz/ml/weka/

collection (abbreviation GTZAN) is the one that was used by George Tzanetakis in previous experiments, presented in (Tzanetakis, 2002). It consists of 1000 pieces of audio equi-distributed among 10 popular music genres. The second collection is the one used in the ISMIR 2004 Rhythm classification contest (ISMIR2004contest), which consists of 698 excerpts of 8 genres from ballroom dance music. The third collection is from the ISMIR 2004 Genre classification contest (ISMIR2004contest) and contains 1458 complete songs, the pieces being unequally distributed over 6 genres. For details about the genres involved in each collection and the numbers of documents in each class we refer to Table 1.

Table 1: Music collections used in genre classification experiments, with genres and number of titles per genre.

GTZAN	**1000**	**ISMIRrhythm**	**698**	**ISMIRgenre**	**1458**
blues	100	ChaChaCha	111	classical	640
classical	100	Jive	60	electronic	229
country	100	Quickstep	82	jazz_blues	52
disco	100	Rumba	98	metal_punk	90
hiphop	100	Samba	86	rock_pop	203
jazz	100	SlowWaltz	110	world	244
metal	100	Tango	86		
pop	100	VienneseWaltz	65		
reggae	100				
rock	100				

4.3 Genre Classification Results

Table 2 presents results of a range of music classification experiments. We state accuracy values on each of the three music collections, i.e. the percentage of music pieces assigned to the correct genre. We can see that the assignment among 10 genres (GTZAN collection) generally performs worse than among the 6 genres of the ISMIRgenre collection. However, the ISMIRrhythm collection delivered in all experiments the best results, which is a great indication that our feature extractor are well-suited to distinguish among different kinds of rhythm. The first row in Table 2 denotes accuracy measures from the Rhythm Patterns feature set as it was implemented at the time of the ISMIR 2004 contest (RP-original). After conducting a wide range of experiments we found a number of optimizations, which led to an improvement of the Rhythm Patterns algorithm (RP-improved). Specifically, the implementation of spectral masking in the feature extraction process has been identified to potentially pose issues to the audio content description, at least regarding specific types of music. The psycho-acoustic transformations involved in the audio feature extraction have been evaluated as crucial for the audio description tasks.

We separately evaluated the performance of the Statistical Spectrum Descriptor and the Rhythm Histogram, which can be obtained from the rows denoted SSD and RH. Subsequently, we investigated the performance of combinations of two or all three feature sets in the machine learning task. Those experiment results are given in the remaining rows of table 2. The final experiment reports an achievement of between 72 % and 84 % classification accuracy, which is an improvement of between 2.5 and 16.4 percentage points over the previous results of the original Rhythm Patterns feature set.

Table 2: Results from music genre classification on 3 music collections with different feature sets and combinations. Accuracy (%).

Feature set	GTZAN	ISMIRrhythm	ISMIRgenre
RP-original	58.5	81.7	71.0
RP-improved	64.4	82.8	75.0
SSD	72.7	54.7	78.5
RH	44.1	79.9	63.2
RP+SSD	72.3	83.5	80.3
RP+RH	64.2	83.7	75.5
SSD+RH	74.9	82.7	79.6
RP+SSD+RH	72.4	84.2	80.0

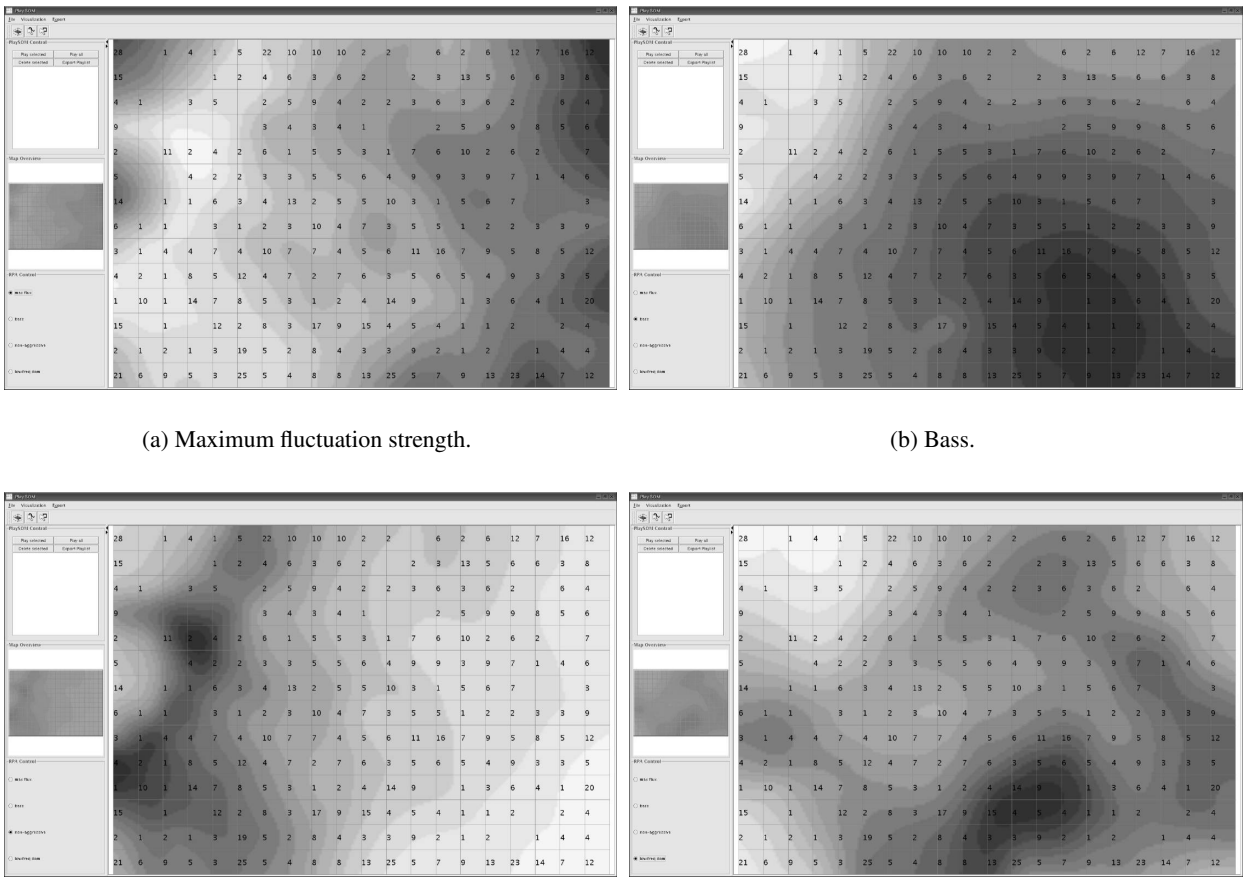
Apart from the classification task, this approach has also been evaluated for pure similarity-based search, evaluating the local stability of the feature set over different 30 sec. segments as well as over different collections, yielding a recall in the range of 50-70% within the top-10 for different segments Leitich and Rauber (2004).

5 APPLICATIONS AND USER INTERFACES

This section presents applications to organize digital music collections based on feature descriptions for audio data like the one presented in the previous sections. Therefore the application of the *SOM* clustering algorithm for mapping music on a two-dimensional map is described and two novel user interfaces are introduced. The experimental results described were obtained by clustering the collection of the ISMIR 2004 Genre classification contest described in Section 4.2, using the improved *Rhythm Patterns* feature set.

5.1 Self-Organizing Map

A range of clustering algorithms can be employed to organize audio by sound similarity based on different feature sets. One model that is particularly suitable, is the *Self-Organizing Map (SOM)*, an unsupervised neural network that provides a mapping from a high-dimensional input space to usually two-dimensional output space (Kohonen, 1982, 2001). During the mapping process topological relations are preserved as faithfully as possible. A *SOM* consists of a set of i units arranged in a two-dimensional grid, each attached to a weight vector $m_i \in \Re^n$. Elements from the high-dimensional input space, referred to as input vectors $x \in \Re^n$, are presented to the *SOM* and the activation of each unit for the presented input vector is calculated using an activation function (the Euclidean distance is a very common activation function). In the next step the weight vector of the unit showing the highest activation (i.e. the smallest Euclidean distance) is selected as the 'winner' and is modified as to more closely resemble the presented input vector. Pragmatically speaking, the weight vector of the winner is moved towards the presented input signal by a certain fraction of the Euclidean distance as indicated by a time-decreasing learning rate α. Consequently, the next time the same input signal is pre-



(a) Maximum fluctuation strength.

(b) Bass.

(c) Non-aggressiveness.

(d) Low frequencies dominant.

Figure 2: Different visualizations of Rhythm Patterns in the *PlaySOM* interface.

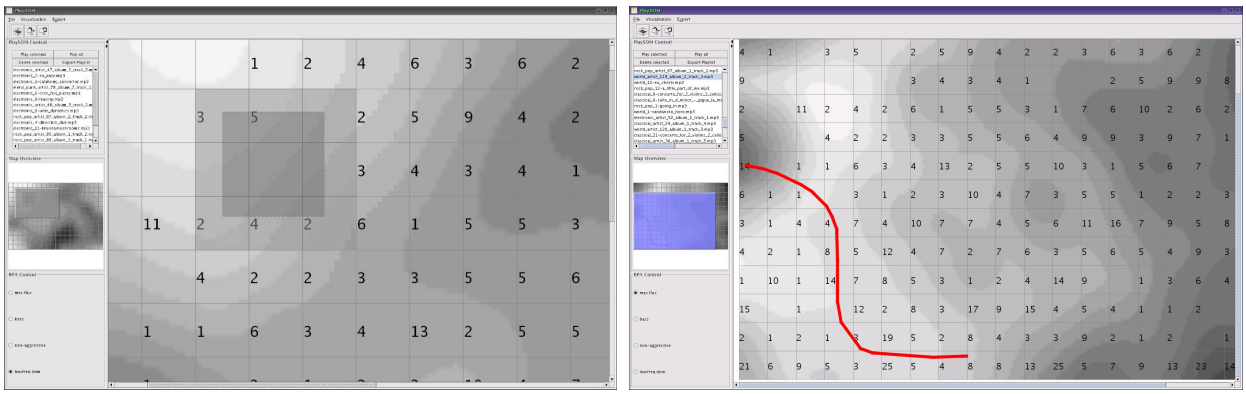
sented, this unit's activation will be even higher. Furthermore, the weight vectors of units neighboring the winner, as described by a time-decreasing neighborhood function, are modified accordingly, yet to a smaller amount as compared to the winner. The result of this learning procedure is a topologically ordered mapping of the presented input signals in two-dimensional space. Accordingly, similar input data are mapped onto neighboring regions of the map. A *SOM* can be trained using all kinds of feature sets, however, in our experiments *Rhythm Patterns* is the obvious choice.

5.2 SOM Visualizations

Due to the fact that the cluster structure of a trained *SOM* is not inherently visible, several visualization techniques have been developed, the most prominent being the *U-Matrix* (Ultsch and Siemon, 1990).

Another useful method that provides insight into the structure of a trained *SOM* is the visualization of component planes, i.e. individual features. Only a single component of the weight vectors is used to color-code the map representation. This information can also be overlayed onto other visualizations using Gradient Fields (Pölzlbauer et al., 2005). In other words, the values of a specific component of the weight vectors are mapped onto a color palette to paint units accordingly allowing to identify regions that are dominated by a specific feature.

Since single component planes do not directly translate into psychoacoustic sensation noticed by the human ear, the *Rhythm Patterns* uses four combinations of component planes according to psychoacoustic characteristics (Pampalk et al., 2002). More precisely, *maximum fluctuation strength* evaluates to the maximum value of all vector components representing music dominated by strong beats. *bass* denotes the aggregation of the values in the lowest two critical bands with a modulation frequency higher than 1 Hz indicating music with bass beats faster than 60 beats per minute. *Non-aggressiveness* takes into account values with a modulation frequency lower than 0.5 Hz of all critical bands except the lowest two. Hence, this feature indicates rather calm songs with slow rhythms. Finally, the ratio of the five lowest and highest critical bands measures in how far *low frequencies dominate*. These characteristics can be used to color the resulting map, providing weather-chart kind of visualizations of the music located in different parts of the map. Figure 2 shows examples for all four kinds of visualizations.



(a) Rectangle selection without preserving track order.

(b) Trajectory selection preserving track order.

Figure 3: The *PlaySOM* interface, its selection models and playlist contents.

5.3 SOM-Based User Interfaces

Two interfaces to digital music collections are described in this section, both are based on the *Self-Organizing Map* clustering algorithm and allow interactive exploration of music collections according to feature similarity of audio tracks. The *PlaySOM* and *PocketSOMPlayer* applications both enable users to explore and browse music collections, select tracks, export playlists as well as listen to the selected songs. The *PlaySOM* is a full interface, offering different selection models, a range of visualizations, advanced playlist refinement, export to external player devices or simply playback of selected songs. The *PocketSOMPlayer*, on the other hand, offers a slim version of the desktop application and is optimized for the *PocketPC* platform, implemented for an iPaq using Java and SWT to ensure platform independency and to be used in a streaming environment.

A mapped music collection visualizing the previously described different *Rhythm Patterns* sub-groups are depicted in Figures 2(a)-(d). When discovering a map of music, the visualizations can provide important clues to the overall organization of a specific map and offer starting points for interactive exploration depending on the characteristics of music one is interested in. For printing purposes we use a linear gray scale comprising 16 colors from dark gray to white representing feature values from low to high (For on-screen use, we emphasize the map metaphor by using a fine-grained color palette ranging from blue via yellow to green reflecting geographical properties similar to the *Islands of Music* (Pampalk et al., 2002)).

The organization of the songs according to the *maximum fluctuation strength* feature is clearly visible in Figure 2(a) where pieces of music having high values are located primarily on the left-hand side of the map. Contrarily, songs with low values are located on the map's right-hand side.

Figure 2(b) shows that the feature *bass* is concentrated on the upper left corner and presents mainly bass-dominated tracks.

The setting for *non-aggressiveness* is depicted in Figure 2(c), the majority of clusters containing high values can be identified on the right-hand side of the map as one would expect regarding the distribution of the *maximum fluctuation strength*, which represents music dominated by strong and fast beats.

Finally, a small cluster where *low frequencies dominate* is located in the upper left of the map as shown in Figure 2(d) and corresponds to the results of *bass* setting, leading to low values in this region.

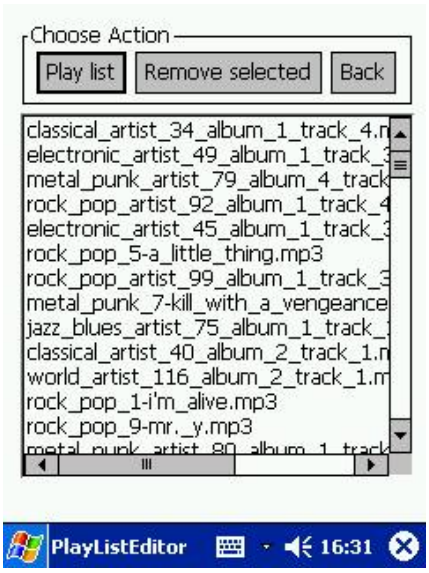
5.4 PlaySOM

The *PlaySOM* application allows users to interact with the map mainly by panning, semantic zooming and selecting of tracks. Users can move across the map, zoom into areas of interest and select songs they want to listen to. They can thereby browse his private collection of a few thousand songs, generating playlists based on track similarity instead of clicking through metadata hierarchies, and either listening to those selected playlists or exporting them for later use. Users can abstract from albums or genres which often leads to rather monotonous playlists often consisting of complete albums or many songs from one genre. This approach enables users to export playlists based on track not on metadata similarity or manual organization.

The *PlaySOM*'s largest part is covered by the interactive map on the right, where squares represent single units of the SOM. Controls for selecting different visualizations and exporting the map data and the current visualization for the *PocketSOMPlayer* are part of the menubar on the top. The left hand side of the user interface contains (1) a playlist of currently selected titles, (2) a birds-eye-view showing which part of the potentially very large map is currently depicted in the main view on the right and (3) controls for the currently selected visualization (as demonstrated by the different settings of the *Rhythm Patterns* in Figure 2).

The icons on the upper left allow the user to switch between the two different selection models and to automatically fit the map to the current screen size. Figure 3 depicts the interaction models that are currently supported

(a) The *PocketSOMPlayer*'s main panel showing a trajectory selection.



(b) *PocketSOMPlayer* user refinement panel.

Figure 4: The *PocketSOMPlayer* interface showing different interaction views.

by the *PlaySOM*. The rectangular selection model allows the user to drag a rectangle and select the songs belonging to units inside that rectangle without preserving any order of the selected tracks. This model is used to select music from one particular cluster or region on the map. Figure 3(a) depicts the selection of a cluster of songs located at the upper left part of the map mainly belonging to the *Electronic* genre, comprising single tracks from *Rock_Pop* and *Metal_Punk* in this example without any specific order. On the other hand, the line selection model allows users to draw trajectories and select all songs belonging to units beneath that trajectory. Figure 3(b) shows a selection of tracks and the according transitions between those genres along the trajectory. The dark region located at the beginning of the trajectory at the left middle of the figure mainly consists of *Electronic* tracks and represents high values in the *maximum fluctuation strength* set of features. Further along the trajectory, the playlist continues with a few more lively and dynamic songs belonging to the *Rock_Pop* and *Metal_Punk* genres, represented by the lighter region, before it turns back to rather tranquil music from the *Classical* genre. In this case the sequence of selected units is of particular importance, because this line chooses a variety of songs according to their position on the map, i.e. their similarity. Hence the line selection model makes it possible to generate playlists including smooth transitions between clusters of tracks. This might be of specific interest when browsing very large music collections or when rather long playlists shall be generated (for example if a playlist for several hours should be generated and several changes in musical style shall occur over time, similar to an *auto-dj* functionality).

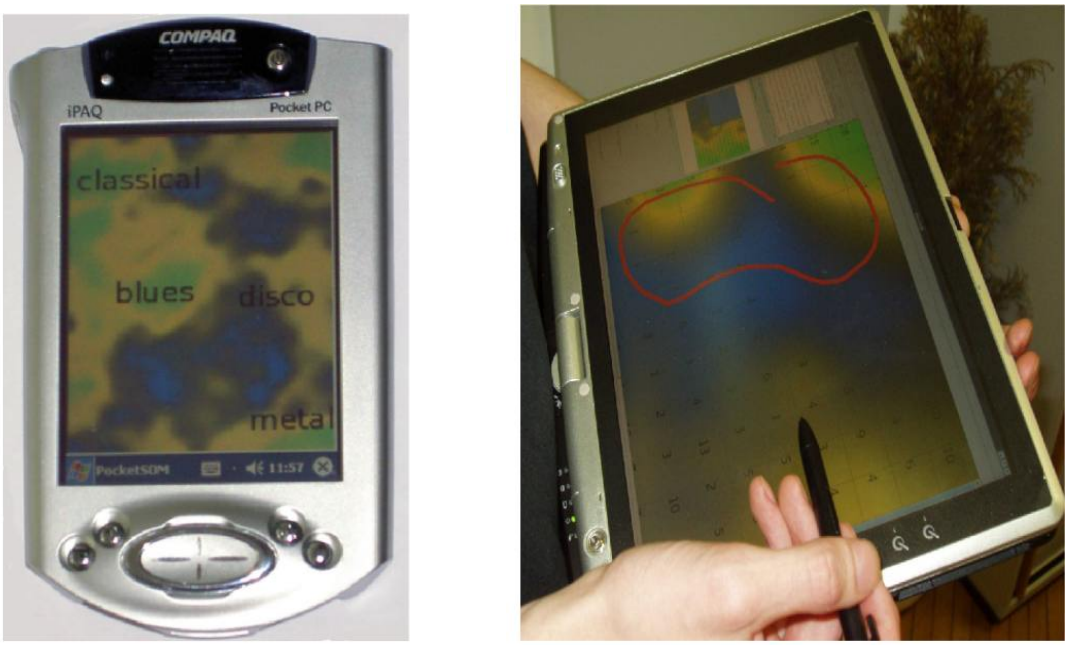
Once a user has selected songs and refined the results by manually dropping single songs from the selection, those playlists can be listened to on-the-fly or exported for later use on the desktop machine or even other platforms like PDAs or Multimedia Jukeboxes if the collection is served via a streaming environment.

Furthermore, the main *PlaySOM* application can easily and efficiently be used on a Tablet PC and used as a touch screen application because of its portable Java implementation (a live demo is shown in 5(b)).

5.5 PocketSOMPlayer

The *PocketSOMPlayer* application offers similar but simplified functionality as the *PlaySOM* being designed for mobile devices such as PDAs or Smartphones. Therefore it only provides the basic functinality of selecting by drawing trajectories and a simplified refinement section, omitting means to zoom or pan the map. Its operational area is likely to be a client in a (wireless) audio streaming environment for entertainment purposes. Regarding the current memory restrictions of PDAs, the use of a streaming server as music repository seems even more appealing than for the desktop application. Nevertheless, the mobile interface could be synchronized with its desktop pendant to take the role of a mobile audio player within the PDA's memory limits.

Figure 4(a) shows the *PocketSOMPlayer*'s main interface, a trajectory selection with an underlying map. Its user refinement view which allows the user to modify the previously selected playlist before listening to the result is depicted in Figure 4(b). (Due to the anonymized format of the ISMIR collection we emphasized on genres instead on individual track names. In real application scenarios, filenames or ID3-tag information would be used for displaying information on the map.) The main panel allows



(a) The *PocketSOMPlayer* application running on an iPaq PDA.

(b) *PocketSOMPlayer* running on a Tablet PC.

Figure 5: Both presented interfaces running on an iPaq and Tablet PC respectively.

the user to draw trajectories and to select the units underneath those trajectories. All songs mapped to the selected units are added to the playlist. The user refinement panel pops up as soon as a selection is finished and provides similar functionality as the *PlaySOM*'s playlist controls, namely the user can delete single songs from the playlist to refine her/his selection. The resulting playlist can then be played, retrieving the MP3s either from the local storage or a streaming server.

Figure 5(a) shows the *PocketSOMPlayer* running on an iPaq PDA without a trajectory selection. The map describes a music repository located on a streaming server running on another machine, accessible via WLAN, in contrast to keeping the music files locally (note that labels are manually assigned to clusters according to the most prominent genres in this example). Selecting tracks via drawing of trajectories on a touch schreen is straightforward, easy to learn and intuitive as opposed to clicking through genre hierarchies and therefore particularly interesting for mobile devices and their handling restrictions.

6 CONCLUSIONS AND FUTURE WORK

We described the extraction of three feature sets for content-based audio description, namely Rhythm Patterns, Statistical Spectrum Descriptor and Rhythm Histogram. The feature sets are used for music similarity tasks such as automatic classification and organization of music collections. The algorithms' performance was evaluated on a music genre classification task. Throughout the experiments, the algorithm for the computation of the Rhythm Patterns has been improved. Together with the two other feature sets, classification accuracy reaches up to 74.9 %, 84.2 % and 80.3 % in three different standard music collections.

The feature sets are applicable to music retrieval tasks (query by similarity), to classification as well as to performing automatic organization tasks.

We presented intuitive visualizations based on the *Self-Organizing Map*, a neural network with unsupervised learning function. For training of the *SOM* we used the automatically extracted feature vectors described before. Furthermore, we presented the *PlaySOM*, a novel user interface to map representations of music collections created by *SOM* clustering. The interface allows user interaction and interactive exploration based on those maps. The *PlaySOM* offers a two-dimensional map with spatial organization of similar tracks and is especially appealing for large or unknown collections, which could hardly be browsed by metadata search only. The application allows users to browse their collections by similarity and therefore find songs similar to ones they know by name in contrast to metadata-based approaches. Moreover, we introduced a PDA application offering similar functionality, showing that alternative approaches to music organization are feasible for mobile devices as well. Both user interfaces are well suited for interactive exploration of collections of digital music because of their different levels of interaction like semantic zooming or on-the-fly playlist generation.

In the future we will further investigate in detail the steps of the computation of audio features for further im-

provement of the content-based description of audio. Future work will also deal with further development of interfaces for mobile devices, especially concentrating on their use in streaming environments. Therefore the combination of such clients with centralized music repositories, offering tighter integration of feature extraction and online exchange of stored information about tracks such as the well known ID3 tags, is going to be evaluated. Moreover, the desktop interface may be extended by more sophisticated methods for playlist generation such as automatic smooth transitions between clusters.

In addition, user studies might be of great help to measure the quality of the *SOM* clustering in combination with the *Rhythm Patterns* feature extraction as the automated quality assessment is very difficult as mentioned before. The current *PlaySOM* implementation is well suited for such studies and on-the-fly evaluation of specific areas on the maps as described in our experiments.

ACKNOWLEDGEMENTS

Part of this work was supported by the European Union in the 6. Framework Program, IST, through the DELOS NoE on Digital Libraries, contract 507618, and the MUSCLE NoE on Multimedia Understanding through Semantics, Computation and Learning, contract 507752.

References

R. B. Dannenberg, B. Thom, and D. Watson. A machine learning approach to musical style recognition. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 344–347, Thessaloniki, Greece, September 25-30 1997.

B. Feiten and S. Günzel. Automatic indexing of a sound database using self-organizing neural nets. *Computer Music Journal*, 18(3):53–65, 1994.

J. T. Foote. Content-based retrieval of music and audio. In *Proceedings of SPIE Multimedia Storage and Archiving Systems II*, volume 3229, pages 138–147, 1997.

ISMIR2004contest. ISMIR 2004 Audio Description Contest. Website, 2004. `http://ismir2004.ismir.net/ISMIR_Contest.html`.

T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.

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

S. Leitich and A. Rauber. Information retrieval in digital libraries of music. In *Proceedings of the 6th Russian Conference on Digital Libraries (RCDL 2004)*, pages 207–215, Pushchino, Russia, September 29 - October 1 2004.

T. Lidy, G. Pölzlbauer, and A. Rauber. Sound re-synthesis from rhythm pattern features: Audible insight into a music feature extraction process. In *Proceedings of the International Computer Music Conference (ICMC 2005)*, Barcelona, Spain, September 5-9 2005.

B. Logan. Content-based playlist generation: Exploratory experiments. In *Proc. 3rd Ann. Symp. on Music Information Retrieval (ISMIR 2002)*, France, 2002.

B. Logan and A. Salomon. A music similarity function based on signal analysis. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, Tokyo, Japan, August 2001.

E. Pampalk, S. Dixon, and G. Widmer. On the evaluation of perceptual similarity measures for music. In *Proceedings of the International Conference on Digital Audio Effects (DAFx-03)*, pages 7–12, London, UK, September 8-11 2003.

E. Pampalk, A. Rauber, and D. Merkl. Content-based organization and visualization of music archives. In *Proceedings of ACM Multimedia 2002*, pages 570–579, Juan-les-Pins, France, December 1-6 2002.

G. Pölzlbauer, M. Dittenbach, and A. Rauber. Visualization technique for self-organizing maps with vector fields to obtain the cluster structure at desired levels of detail. In *Proceedings of the International Joint Conference on Neural Networks, IJCNN*, Montreal, Quebec, July 31-August 5 2005.

A. Rauber and M. Frühwirth. Automatically analyzing and organizing music archives. In *Proceedings of the European Conference on Research and Advanced Technology for Digital Libraries (ECDL)*, Darmstadt, Germany, September 4-8 2001.

A. Rauber, E. Pampalk, and D. Merkl. Using psychoacoustic models and self-organizing maps to create a hierarchical structuring of music by musical styles. In *Proceedings of the International Conference on Music Information Retrieval*, pages 71–80, Paris, France, October 13-17 2002.

A. Rauber, E. Pampalk, and D. Merkl. The SOM-enhanced JukeBox: Organization and visualization of music collections based on perceptual models. *Journal of New Music Research*, 32(2):193–210, June 2003.

C. Spevak and E. Favreau. Soundspotter - a prototype system for content-based audio retrieval. In *Proceedings of the 5. International Conference on Digital Audio Effects (DAFx-02)*, Hamburg, Germany, September 26-28 2002.

M. Torrens, P. Hertzog, and J. L. Arcos. Visualizing and exploring personal music libraries. In *ISMIR 2004, User Interfaces*, pages 421–424, Barcelona, Spain, October 2004.

G. Tzanetakis and P. Cook. Marsyas: A framework for audio analysis. *Organized Sound*, 4(30), 2000.

G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, July 2002.

G. Tzanetakis. *Manipulation, Analysis and Retrieval Systems for Audio Signals*. PhD thesis, Computer Science Department, Princeton University, 2002.

A. Ultsch and H. Siemon. Kohonen's self-organizing feature maps for exploratory data analysis. In *Proceedings of the International Neural Network Conference*

(INNC'90), pages 305–308, Dordrecht, Netherlands, 1990.

F. Vignoli, R. van Gulik, and H. van de Wetering. Mapping music in the palm of your hand, explore and discover your collection. In *ISMIR 2004, User Interfaces*, pages 409–414, Barcelona, Spain, October 2004.

E. Wold, T. Blum, D. Keislar, and J. Wheaton. Content-based classification search and retrieval of audio. *IEEE Multimedia*, 3(3):27–36, Fall 1996.

H. Zhang and D. Zhong. A scheme for visual feature based image indexing. In *Proceedings of the IS&T/SPIE Conference on Storage and Retrieval for Image and Video Databases*, pages 36–46, San Jose, CA, February 4–10 1995.

E. Zwicker and H. Fastl. *Psychoacoustics - Facts and Models*, volume 22 of *Springer Series of Information Sciences*. Springer, Berlin, 1999.