Hierarchical Automatic Audio Signal Classification*

JUAN JOSÉ BURRED

Communication Systems Group, Technical University Berlin, Berlin, Germany

AND

ALEXANDER LERCH, *AES Member*

zplane.development, Berlin, Germany

The design, implementation, and evaluation of a system for automatic audio signal classification is presented. The signals are classified according to audio type, differentiating between three speech classes, 13 musical genres, and background noise. A large number of audio features are evaluated for their suitability in such a classification task, including MPEG-7 descriptors and several new features. The selection of the features is carried out systematically with regard to their robustness to noise and bandwidth changes, as well as to their ability to distinguish a given set of audio types. Direct and hierarchical approaches for the feature selection and for the classification are evaluated and compared.

0 INTRODUCTION

Over the last decade the exponential growth of the Internet has made huge amounts of information easily available to millions of people. Furthermore, advances in networking technologies, as well as in coding and compression algorithms, have brought about increased bandwidth while allowing to make optimal use of it. As a result, content requiring high levels of bandwidth, such as video and audio streams or any kind of multimedia documents, coexists nowadays with the traditional textual and graphical data. It is not unlikely that, in the near future, music available on the Internet will overtake the usual distribution of audio stored on CDs or DVDs. This new way of producing, sharing, and storing information is beginning to set unprecedented commercial and legal paradigms.

However, although the variety and quantity of accessible data are enormous, the way they can be managed and searched for is frustratingly limited. At this point it is only possible to search the Internet using textual queries, for example, using a Web browser. When searching for an image, a video, or an audio clip, one must rely on the textual information manually attached at one time to the corresponding file. This information is often inaccurate, and in most cases it consists of only the title and the author or performer. In many cases there is no such information at all, the file name being the only hint about its content. Today it is not possible to perform such natural actions as searching for an image containing certain objects, a movie scene featuring a given actor, or organizing a collection of music files according to their genre or mood, unless this information has been attached manually to the file beforehand, which in most cases is a nearly unfeasible task.

These considerations led to the fast growth of the content analysis (CA) or machine perception research field in recent years. Its goal is to make computers capable of automatically extracting metadata, that is, information about the content from data. In particular, audio content analysis (ACA), also called computer audition or machine listening, deals with the extraction of information from sounds.

The term information retrieval (IR) is often used as a synonym for content analysis, in the sense that content information is retrieved from the signals. However, to retrieve information could also mean to search for a given entry in a database, and early IR fields include, for example, the study of text-based query techniques such as the ones used to browse information on the Internet. In order to avoid misunderstandings, the clearly defined term of content analysis will be used here.

*Manuscript received 2003 October 15; revised 2004 April 20.

One of the possible applications of ACA is classification. In a classification system the input signal is analyzed and a label describing that signal is delivered at the output. There are many possible criteria upon which the signals can be separated, such as timbral or melodic content, pitch, or musical tempo. In the context of multimedia data managing and browsing, a very useful classification is the one that assigns labels describing the type or category of the signal, that is, whether the analyzed sound is speech or noise, whether it belongs to a certain music genre, and so on. This is the natural way to classify audio signals, and all common sound databases on the Internet, sound archives, as well as retail shops are organized this way.

The following possible applications of a category-based audio classification show why this research field is so active at present.

- *Audio database indexing* A stored audio collection is scanned, assigning an audio class label to each file. This enables later audio type-based browsing, such as searching a database for specific musical genres.
- *Intelligent signal processing* Possibilities include automatic equalization and automatic control of dynamics processing, such as type-dependent adaptation of loudness in broadcasting.
- *Automatic bandwidth allocation* A communication network with audio classification capabilities could dynamically allocate bandwidth for the signal being transmitted. More bandwidth would be allocated for music than for speech transmissions, and no bandwidth at all if only background noise was detected. This would help multiplexing systems to work more efficiently.
- *Segmentation* The signal to be classified can contain a sequence of different audio classes, such as alternating speech, music, and background in a soundtrack. Segmentation consists of finding the exact transition moment between two consecutive audio types. This allows to extract desired parts from a stream and to discard undesired ones.
- *Intelligent audio coding* Some audio coding algorithms are designed to work in an optimal way with speech signals, others with music signals. Audio classification would allow to switch automatically between coders, or to select suitable encoder settings, depending on the audio type at the input.
- *Broadcast browsing* A classification system with real-time capabilities could allow to scan a radio or TV dial in order to find the desired music genre.

1 RELATED WORK

Automatic speech recognition (ASR) is by far the field that has gathered the most attention of the audio content analysis community over the last 20 years. ASR systems can be regarded as a particular case of categorical audio classification, in which a given spoken fragment (be it a sentence, a word, or a phoneme) is classified into a possible textual transcription. ASR has raised great interest among researchers and investors, because it offers a natural human–machine interaction. Less interest has been paid to nonspeech audio classifiers.

But, as mentioned, this situation has been changing rapidly with the advent of audio material available on the Internet. In recent years many research projects have started with the goal of developing and evaluating classifiers that take into account not only speech, but also other sound signals, such as music and environmental noise [1] [2]. Classifiers can be broadly divided into taxonomic classifiers and clustering classifiers. Taxonomic classifiers make use of so-called supervised learning techniques and allow the implementation of a given category scheme (or taxonomy) defined beforehand by the user or by the implementation itself. In contrast, in clustering systems the separation into categories is up to the algorithm, which groups the audio samples according to some measure of similarity, such as timbral similarity. In this section only taxonomic classifiers will be considered. The interested reader may consult the references about audio clustering systems [3]–[6].

A system may also be characterized according to its real-time capabilities. Real-time classifiers are capable of updating the classification results at intervals usually in the range of milliseconds. They are useful in real-time applications in which the input signal consists of a sequence of different types of audio, and which need an immediate update of the class detection. This is the case, for example, for automatic dynamic control in broadcasting or automatic bandwidth allocation in a transmission line. Non-real-time classifiers analyze a longer fragment of the signal before they provide a classification result. They are usually more accurate than real-time classifiers, since long-term characteristics, which can play a key role in describing a signal, can be measured in this case. The most common non-real-time application is the automatic indexing of audio databases.

All automatic classification systems rely on the techniques provided by the science of pattern recognition (PR). The first step in any PR application is to extract a set of measures, or features, from the signal to be classified. Each of these features constitutes one element of the feature vector that will represent the signal in the so-called feature space, which will have as many dimensions as extracted features. In taxonomic systems a set of sample feature vectors representing each of the audio classes is used to train a statistical or neural classifier. The classifier infers some kind of decision rules that will be applied to assign a class to an incoming unknown signal.

Some systems are capable of extracting the features in real time. The real-time feature extraction should not be confused with the aforementioned real-time classification. A real-time classifier does need the features to be computed in real time, but a non-real-time system could be able to extract features in real time while being unable to perform an immediate classification.

One of the first general audio classification systems was developed by the Muscle Fish company in 1996 [7]. In this system there are no prespecified classes since the training is up to the user. Pitch, brightness modeled by the spectral centroid, spread, and mel frequency cepstral coefficients (MFCC) are used as features (see Section 4). The system uses a simple Gaussian (GS) classifier (see Section 6), and its main application is similarity retrieval. It has been tested using short, individual sound segments such as sound effects and musical instrument notes, which train very specific classes such as laughter, animals, bells, crowds, or water.

A system proposed by Scheirer and Slaney [8] uses 13 features such as rolloff, centroid, flux, zero crossings, and beat features to distinguish between speech and music. Four different classifiers were evaluated: a GS classifier, a Gaussian mixture model (GMM), and two variants of the *k* nearest neighbor (*k*-NN) classifier (see Section 6). The four provided very similar classification performance. A final classification rate of 98.6% is reported. When using the system as a three-way classifier to separate music, speech, and simultaneous speech and music, the rate drops to 65%.

Foote [9] proposed a tree-based quantizer as the classifier, which partitions the feature space into regions with feature vectors belonging to maximally different classes. MFCCs are used as features. A three-way classification between speech, music, and nonvocal sound is implemented using this technique. No specific classification accuracy measures are provided in his paper.

Zhang and Kuo [10], [11] have developed a system that performs classification in two levels. At the first level, which they call the coarse level, sound is classified into speech, music, environmental sound, and silence. At the second-stage or fine-level classification, sounds are further divided into finer classes. The work focuses on the classification of environmental sounds into 10 classes such as applause, birds, laugh, or rain. At this level an accuracy of 80% is reported.

Pye [12] addresses the classification of audio signals compressed in MP3 format. Two approaches are compared with respect to performance and computational cost. In the first, MFCCs are used as features, which requires a previous MP3 decompression. The other proposed method consists of deriving an MFCC-like set of features performing only a partial decompression, and is called MP3CEP. GMM and a tree-based vector quantizer such as the one used by Foote [9] are the evaluated classifiers. Music classification is performed into the following six classes: blues, easy listening, classical, opera, dance (techno), and indie rock. The best classification rate for both MFCC and MP3CEP was obtained using the GMM classifier. While MFCCs performed slightly better (92%) than MP3CEP (90.9%), in the latter case the computation was more than five times faster.

The system developed by Casey [13], [14] is intended for audio classification and retrieval in a generalized way, that is, the definition and training of all classes is made by the user. The features are obtained from a compact representation of the spectrum using singular value decomposition (SVD). This system uses a hidden Markov model (HMM)–based classifier and has been adopted by the MPEG-7 standard.

In Tzanetakis and coworkers [15], [16] music signals are divided into classical, country, disco, hip-hop, jazz, rock, blues, reggae, pop, and metal with an accuracy of 61%. Classical music is further divided into four subgenres (88% accuracy), and jazz into six subgenres (68%). The musical taxonomy is extended by introducing a previous music/speech discriminator. Speech signals are further divided into male speech, female speech, and speech with noisy background (74%). The system uses 30 different timbral, rhythmic, and pitch-related features.

In Lu and Jiang [17] audio is labeled as speech, music, environment sound, and silence, achieving an accuracy of 96.51%. Classification is performed in two steps. At the first step it is distinguished between speech and nonspeech signals using a combination of a *k*-NN classifier based on zero crossings, flux, and energy features, and a set of heuristic rules based on a linear-prediction-related feature called linear spectral pairs (LSPs). At the second stage nonspeech is classified into music, environment, or silence by a set of heuristic rules based on flux, the ratio of noisy frames, and a periodicity measure based on the autocorrelation, similar to the harmonic ratio defined in MPEG-7 (see Section 4.2).

One proof of the increasing significance of content analysis is that there already exists an international standard defining a set of techniques for analyzing and describing raw data. Unlike the previously published standards of the Moving Picture Experts Group (MPEG-1, MPEG-2, and MPEG-4), the new MPEG-7 standard [14], [18] does not deal with the compression of signals, but with its content-based description.

The main goal of MPEG-7 is to standardize feature extraction for any type of digital data and to define a language that describes these data in terms of their features. In the context of the standard, a feature is called a low-level descriptor (LLD). In the present work four audio LLDs have been selected for implementation and evaluation in the classification task.

2 SCOPE AND OVERVIEW

Although many combinations of features and classifiers have been evaluated in the publications mentioned, little attention has been paid to the following issues:

- *Genre dependency of features* Some features are more suitable than others when classifying into a given set of subgenres. For instance, a measure of beat strength is more likely to perform better in separating classical from pop music than in classifying into chamber music subgenres.
- *Problems of dimensionality* In PR applications, adding new features (that is, adding new dimensions in the feature space) does not necessarily result in a higher classification accuracy. Reducing the number of features allows to reduce computational costs while maintaining a similar classification accuracy. In some cases the classification rate can even benefit from this reduction in dimensionality.
- *Systematic feature selection* There is a wide variety of automatic feature selection algorithms in the PR literature that have not yet been applied in the context of audio classification.
- *Inappropriate taxonomies* Many previously proposed musical taxonomies are too simple or musicologically inconsistent.

In this work the preceding issues have been investigated and applied in the design and implementation of a prototype application for audio signal classification. A common approach in audio classification is to use all features in a single-stage classification, in which the audio class is decided directly out of all possible final classes (flat or direct approach). However, the genre dependency of the features suggests a hierarchical classification scheme in which, at each step of the classification, only the features that are most appropriate to distinguish between the corresponding subclasses are used (hierarchical approach). Besides allowing to account for this genre dependency, a hierarchical system also has several other advantages, such as the ability to reduce the probability of costly errors and easy expansion capabilities, which will be addressed in detail in Section 6. The main motivation of this work was to compare the classification accuracy of both direct and hierarchical approaches.

The system presented has been designed to operate on audio signals stored as audio files. Furthermore, the files are supposed to be homogeneous, that is, to contain only one type of audio. Hence a constant classification update is not needed in this case, and long-term sections of the signals can be analyzed, that is, our system is a non-real-time classifier. Since a fixed, previously defined audio taxonomy is to be implemented, it is also a taxonomic classifier.

The first step in the design of the system consists of determining the classes into which the signals are going to be classified. The choice of audio taxonomy used is addressed in Section 3. In Section 4 a large number of audio features used for classification purposes are reviewed. They include well-known physical and perceptual features, audio descriptors defined in the MPEG-7 standard, as well as new features proposed here. All of them have been implemented and evaluated. The problem of high dimensionality is addressed by systematically reducing the number of features by means of a robustness test and a hierarchical feature subset selection algorithm (Section 5.2).

Finally a density estimation classifier (Gaussian mixture model) and a nonparametric classifier (*k*-nearest neighbor), both in their direct and hierarchical variants, are thoroughly evaluated with respect to classification accuracy and computational performance (Sections 6 and 7). The influence of the length of the classification intervals has also been examined.

In contrast to previous work, the final version of the present system is fully hierarchical, both in the feature selection and in the classification itself. A preliminary version of the system was presented in our previous work [19]. Similar principles have been applied by Peeters and Rodet [20] to the problem of musical instrument detection.

3 AUDIO TAXONOMY

Some of the audio taxonomies proposed in related work may serve well in demonstrating the system capabilities, but are not likely to be useful in a final user-oriented application, since they are often incomplete or musicologically inconsistent. An example of incompleteness is the absence of classical music in Lambrou et al. [21] or Casey [13]. Musicologically inconsistent is, for example, the distinction between classical and opera or the choice of indie rock or metal instead of the more general category of rock in Pye [12] and Tzanetakis and Cook [16], respectively. Another incoherence is the fact that, at the same classification level, some classes are defined according to their musical genre and others according to their instrumentation, such as the piano class in Lambrou et al. [21] or the quartet and piano jazz subgenres in Tzanetakis and Cook [16].

For these reasons a special effort was made to find an audio taxonomy that would be at the same time:

- Complete enough to allow an acceptable classification of as much input signal types as possible
- Musicologically consistent to avoid ill-defined classes
- Simple enough to allow class separation by feasible features.

After studying several possibilities, the taxonomy shown in Fig. 1 was defined, which contains a total number of 17 classes (three speech classes, 13 music classes, and one background noise class). At the first level of the taxonomy, audio is recognized as speech, music, or background noise. Speech and music signals are then classified into more specific classes, such as male and female speech, orchestral and chamber music, rock, pop, and jazz. It should be noted that the taxonomy was designed following only musicological considerations and was determined before the features were proposed (that is, the classification scheme is user-oriented rather than feature-oriented).

Classical music was divided according to instrumentation, and nonclassical music according to style. This is not only because it is a common practice in many market-related taxonomies, but also because to divide classical music accurately according to historical periods such as baroque, romantic, and so on, would require very sophisticated features which are still nonfeasible.

Some examples of genres that are not intended to fit into this work's taxonomy are the following: a capella choral music, wind bands, electronic contemporary classical music, nonwestern classical music (such as Indian or Chinese classical music), traditional and ethnic music, and other western popular genres such as funk, reggae, ska, country, and soul. Of course, these genres can constitute the basis for future expansions of the taxonomy.

50 audio examples were collected for each of the 17 classes, resulting in an 850-file database. The samples were obtained from CDs, uncompressed MP3 files, and radio broadcasts, so that the database represented a wide range of audio quality levels. Each sample consists of a representative excerpt of around 30 seconds, which seems an appropriate length both to allow the evaluation of different classification intervals and to contain enough distinguishing characteristics of the classes.

4 FEATURE EXTRACTION

Virtually all audio features are extracted by breaking the input signal into a succession of analysis windows or frames, each of around 10–40-ms length, and computing one feature value for each of the windows. One possible approach is to take the values of all features for a given analysis window to form the feature vector for the classification decision, which allows to obtain class assignments almost in real time, thus obtaining a real-time classifier.

To achieve better classification results, the concept of texture window has been introduced in previous work [16]. It allows to extract long-term characteristics of the signal and to measure the variation in time of each feature, which very often provides a better description of the signal than the feature itself. A texture window is a long-term segment (in the range of seconds) containing a number of analysis windows. In the texture-based approach only one feature vector for each texture window is generated. The features are not directly the values obtained in each analysis window, but statistical measures of the values obtained for all analysis windows within the current texture window. In this case real-time classification is not possible, since at least one whole texture window has to be processed to obtain a class decision. As mentioned, this is the approach used in the present system. In particular, for each underlying frame-based feature the four following texture-based subfeatures are computed: mean (M), standard deviation (S), mean of the derivative (DM), and standard deviation of the derivative (DS) over all analysis frames within a texture window.

Since the analyzed audio files are supposed to contain only one type of audio, a single class decision is made for each one, which can be derived following one of two possible approaches. The first approach, which we will call single vector mode, consists of taking the whole file length as the texture window. In this way, each file is represented by a single feature vector, which in turn is subjected only once to classification. The other approach, the texture window mode, consists of defining shorter texture windows and making several class decisions along each file, one for each texture window. At the end of the file the decisions are averaged to obtain a final class decision. The average computation is weighted by the certainty of each class decision.

Before proceeding with the feature extraction, the input signal is downmixed to a mono signal if it has more than one channel, and normalized so that its maximum absolute amplitude equals unity.

In the following equations the subindex r indicates the number of the current frame. In this way $x_r[n]$ denotes the frame in the time domain, where n is the time index, and $X_r[k]$ denotes the short-time Fourier transform (STFT) of that frame, where k is the frequency coefficient or bin index.

4.1 Timbral Features

The following well-known timbral features have been implemented and evaluated:

- *Zero crossings* The zero crossings (zc) feature [8], [16] counts the number of times that the signal amplitude changes signs in the time domain during one frame x_r of length N,

$$\mathrm{ZC}_r = \frac{1}{2}\sum_{n=1}^{N} |\mathrm{sign}(x_r[n]) - \mathrm{sign}(x_r[n-1])| \qquad (1)$$

where the sign function is defined by

$$\mathrm{sign}(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases}.$$

For single-voiced signals, zero crossings have been used in the literature to make a rough estimation of the fundamental frequency. For complex signals it is a simple measure of noisiness.

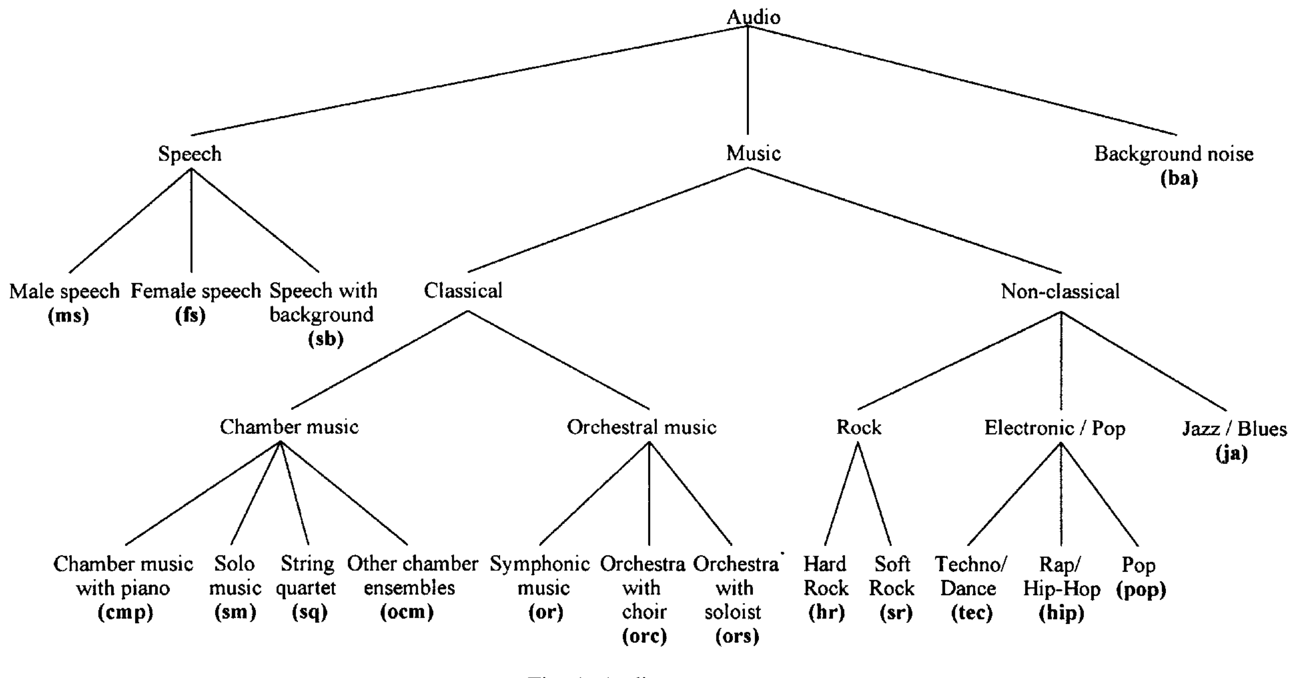


Fig. 1. Audio taxonomy.

- *Centroid,* or gravity center of the spectrum

$$C_r = \frac{\sum_{k=1}^{N/2} f[k]\,|X_r[k]|}{\sum_{k=1}^{N/2} |X_r[k]|} \tag{2}$$

where N is the number of FFT points, $X_r[k]$ is the STFT of frame x_r, and $f[k]$ is the frequency at bin k. Centroid models the sound sharpness. Sharpness is related to the high-frequency content of the spectrum, since higher centroid values correspond to spectra skewed to the range of high frequencies. Due to its effectiveness to describe spectral shape, centroid measures have frequently been used in audio classification tasks [7], [8], [16].

- *Rolloff* Another measure of spectral shape which was first proposed as a feature to distinguish between voiced and unvoiced speech [18], [16]. Like the centroid, it is also a measure of spectral shape and yields higher values for right-skewed spectra. In fact, both features are strongly correlated. The rolloff is defined as the frequency below which 85% of the accumulated magnitudes of the spectrum is concentrated. That is, if K is the largest bin that fulfills

$$\sum_{k=1}^{K} |X_r[k]| \le 0.85 \sum_{k=1}^{N/2} |X_r[k]| \tag{3}$$

then the rolloff is $R_r = f[K]$.

- *Flux* A measure of the spectral rate of change, which is given by the sum across one analysis window of the squared difference between the magnitude spectra corresponding to successive signal frames,

$$F_r = \sum_{k=1}^{N/2} (|X_r[k]| - |X_{r-1}[k]|)^2. \tag{4}$$

Flux has been found to be a suitable feature for the separation of music from speech, yielding higher values for music examples [8], [16].

- *Mel frequency cepstral coefficients* MFCCs are a compact representation of the spectrum of an audio signal taking into account the nonlinear human perception of pitch, as described by the mel scale. They are one of the most used features in speech recognition and have recently been proposed to analyze musical signals as well [9], [16]. A recent study by Logan [22] confirmed that MFCCs are appropriate to represent musical signals. MFCCs are computed by grouping the STFT coefficients of each frame into a set of 40 coefficients, using a set of 40 weighting curves that simulate the frequency perception of the human hearing system. Then the logarithm of the coefficients is taken, and a discrete cosine transform (DCT) is applied to decorrelate them. Only the five first coefficients have been taken as features, which proved to work efficiently [16].

4.2 MPEG-7 Features

There are several LLDs intended for single-voiced, quasi-periodic sound segments, which have not been implemented here since they would produce unreliable results with general, complex signals. Only MPEG-7 audio descriptors that are suitable for any audio type have been taken into consideration. The following descriptors were implemented:

- *Audio spectrum centroid (ASC)* A perceptually adapted definition of the centroid, which introduces a logarithmic frequency scaling centered at 1 kHz,

$$\mathrm{ASC}_r = \frac{\sum_{k=1}^{N/2} \log_2(f[k]/1000)\,P_r[k]}{\sum_{k=1}^{N/2} P_r[k]} \tag{5}$$

where P_r is the power spectrum of frame r.

- *Audio spectrum spread (ASS)* It describes how the spectrum is concentrated around the centroid and is defined as

$$\mathrm{ASS}_r = \sqrt{\frac{\sum_{k=1}^{N/2} [\log_2(f[k]/1000) - \mathrm{ASC}_r]^2\,P_r[k]}{\sum_{k=1}^{N/2} P_r[k]}} \tag{6}$$

where ASC_r is the centroid defined by Eq. (5). Low spread values indicate that the spectrum is highly concentrated near the centroid; high values mean that it is distributed across a wider range at both sides of the centroid.

- *Audio spectrum flatness (ASF)* A measure of the deviation of the spectral form from that of a flat spectrum. Flat spectra correspond to noise or impulse-like signals; thus high flatness values indicate noisiness. Low flatness values generally indicate the presence of harmonic components. Instead of calculating one flatness value for the whole spectrum, a separation in frequency bands is performed, resulting in one vector of flatness values per time frame. The flatness of a band b is defined as the ratio of the geometric and the arithmetic means of the power spectrum coefficients within that band,

$$\mathrm{ASF}_r[b] = \frac{\sqrt[ih[b]-il[b]+1]{\prod_{k=il[b]}^{ih[b]} P_r[k]}}{\frac{1}{ih[b]-il[b]+1}\sum_{k=il[b]}^{ih[b]} P_r[k]} \tag{7}$$

where $il[b]$ and $ih[b]$ denote the first and last frequency bins for that band, respectively. In this work each vector has been reduced to a scalar by computing the mean value across the bands for each given frame, thus obtaining a scalar feature that describes the overall flatness.

- *Harmonic ratio (HR)* A measure of the proportion of harmonic components within the spectrum, defined as the maximum value of the autocorrelation (AC) of each frame,

$$\mathrm{HR}_r = \max_{\tau=1,\,N-1} \{R_r[\tau]\} \tag{8}$$

where $R_r[\tau]$ is the autocorrelation of frame r at lag τ and N is the number of samples in each frame.

It should be noted that the standard defines additional steps concerning the grouping of coefficients which have been omitted here for the sake of simplicity. For complete, detailed specifications see [23].

4.2.1 Modifications to MPEG-7 Harmonic Ratio

When implementing the MPEG-7 harmonic ratio, some inconsistencies were observed in the standard. As can be seen from Eq. (8), values near the main peak of the autocorrelation (at $\tau = 0$) are not skipped when searching for the maximum, which results in harmonic ratio values very close to 1 for virtually all audio classes. This fact is illustrated in Fig. 2, where it can be observed that the first peak of the AC will most likely be the highest. This is especially the case for complex signals where, although there can be other peaks denoting periodicities, they are most likely to have less amplitude than the peak corresponding to little lags. For this reason the search indices have been modified in the present implementation in such a way that the first peak is ignored. Also, the rightmost search limit of $N - 1$ has been replaced by K, which is the lag corresponding to the minimum fundamental frequency expected $f_{0,\mathrm{min}}$ ($K = f_s/f_{0,\mathrm{min}}$, where f_s is the sampling frequency). As it will be shown, this modified version proved to work better in the classification task.

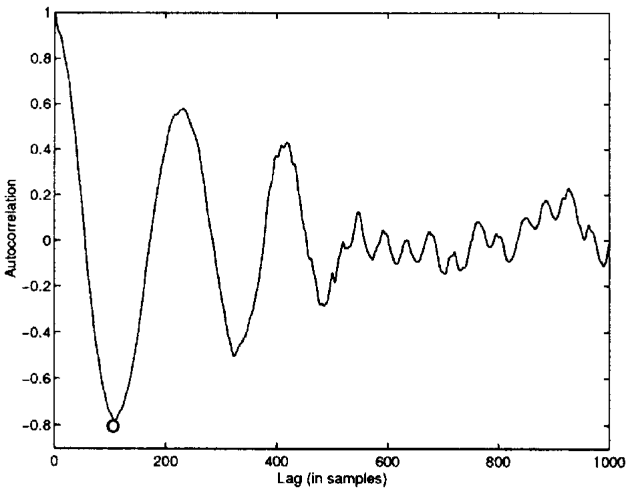


Fig. 2. Autocorrelation of a frame.

4.3 Rhythm Features

Just measuring the tempo in beats per minute (bpm) is not interesting in the context of genre classification, since different pieces belonging to the same genre can have very different tempo properties, whereas pieces from different genres can have the same tempo. For classification purposes it is more interesting to extract information about the rhythmical structure and beat strength. An example is the regularity of the beats, which is expected to be higher in rock and pop examples than in the majority of classical excerpts, where deviations from the original tempo, such as ritardandos or rubato sections, are common. Beat strength also seems to be a valuable feature. For instance, it is likely to be higher in techno music than in jazz. A beat histogram is a curve describing beat strength as a function of a range of tempo values, and allows the extraction of the properties mentioned. Peaks on the histogram correspond to the main beat and other subbeats. Several methods have been proposed for its computation [16], [24]. In this work a method similar to the one presented by Scheirer [24] has been used. Details about the implementation can be found in [25].

The result is a curve describing beat strength as a function of the bpm values. Fig. 3(a) shows the beat histogram of a hip-hop excerpt. The high peaks denote a high overall beat strength. The main peak is at 90 bpm and the second at 45 bpm, denoting a high rhythmic regularity. In contrast, the jazz example in Fig. 3(b) shows a significantly

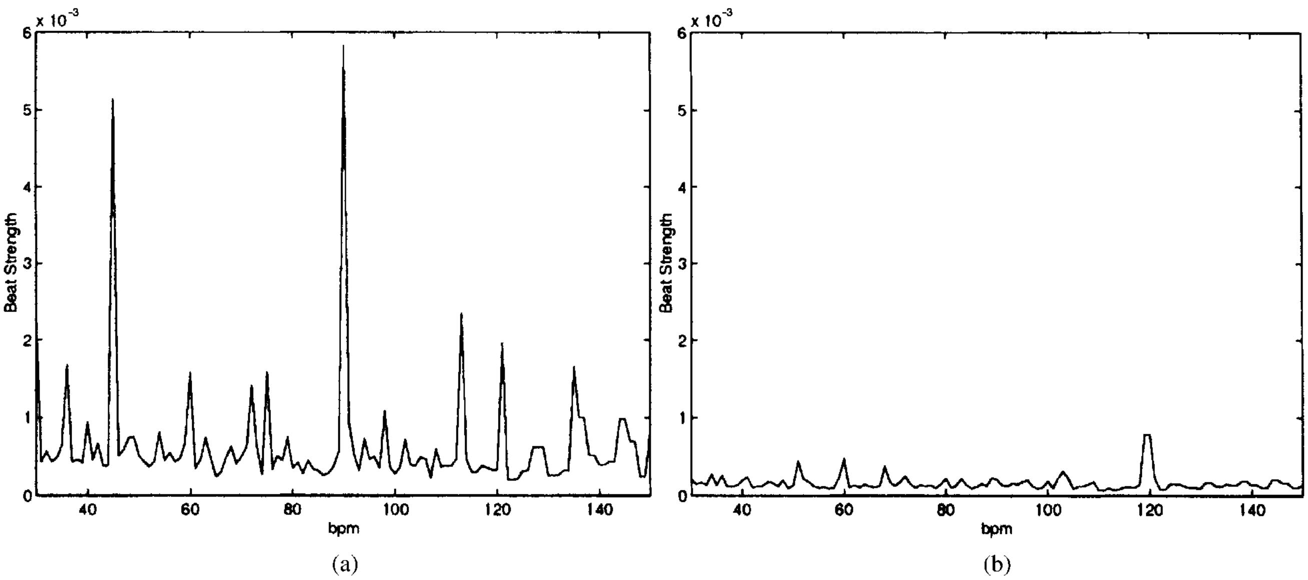


Fig. 3. Beat histogram examples. High peaks correspond to high beat strength; peaks separated by integer bpm multiples denote rhythm regularity. (a) Hip-hop. (b) Jazz.

lower beat strength, but it is still possible to distinguish the main peak at 120 bpm. Many classical examples have irregular beat histograms in which no main peaks and no clear regularity factors can be detected.

All rhythm features were extracted from the beat histograms as follows:

- *Beat strength* To obtain an overall measure of beat strength, the following statistical measures of the histogram have been evaluated: mean, standard deviation, mean of the derivative, standard deviation of the derivative, third- and fourth-order central moments (called skewness and kurtosis, respectively), and entropy. These measures are computed in the "beat domain" and should not be confused with the time-based statistical subfeatures.
- *Rhythmic regularity* A beat histogram in which the peaks are spaced periodically denotes high rhythmic regularity. This can be measured by the normalized autocorrelation function of the beat histogram. It will contain clear peaks for rhythmically regular music examples, and will be the more linear the weaker the regularity is, as shown in Fig. 4. To reduce this to a scalar measure of rhythm regularity, the mean across the lags of the difference between the autocorrelations and the linear function depicted in the figure is computed.

It should be noted that although the computation is performed on a frame-by-frame basis, histograms are obtained in long-term intervals given by the texture windows. Hence all of the features related to the beat histogram are single-valued features to which the time-domain mean and standard deviation subfeatures are not applicable.

4.4 Other Features

The features grouped in this last section describe the signal regarding its dynamic properties, its statistical behavior, and its predictability.

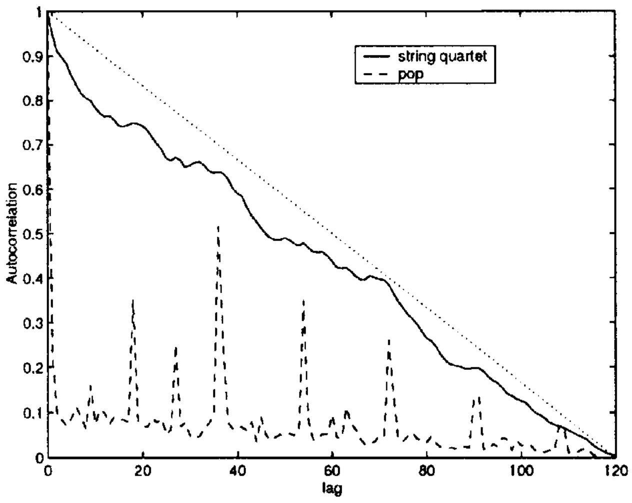


Fig. 4. Illustration of rhythmic regularity feature. Beat histogram autocorrelation from pop and string quartet and reference linear function.

- *Root mean square* Rms energy of each signal frame,

$$\mathrm{rms}_r = \sqrt{\frac{1}{N}\sum_{n=1}^{N} |x_r[n]|^2}\,. \tag{9}$$

- *Time envelope* Maximum of each frame's absolute amplitude.
- *Low energy rate* Percentage of frames within a file that have an rms energy lower than the mean rms energy across that file. It should be noted that apart from the beat-histogram-based features, this is the only feature that is not computed on a frame-by-frame basis, but on a texture window basis. Therefore the statistical subfeatures are not applicable in this case either.
- *Loudness* The previous dynamic-related features are based on physical measures such as amplitude or energy. A better adaptation to the human perception of sound dynamics is provided by the measurement of loudness. A basic exponential model of loudness of the form

$$L_r = E_r^{\alpha} \tag{10}$$

has been used, where E_r is the energy of the current frame. An exponent value of $\alpha = 0.23$ proved to be adequate in the case of noise, and is also likely to be applicable to other broad-band sounds such as music [26], [27]. This model proved to be highly effective in spite of its simplicity.
- *Central moments* The third- and fourth-order central moments of the time-domain audio signal, that is, its skewness and its kurtosis, are evaluated as possible audio features.
- *Predictability ratio* Ratio of the energy of the linearly predicted signal to the energy of the original signal. A linear prediction of the 12th order was used.

5 FEATURE SELECTION

In total, 20 frame-based features, plus one texture-window-based feature (low energy rate) and nine beat-histogram-based features have been evaluated. Furthermore, the four subfeatures, which are applicable to all frame-based features, make a total number of 90 dimensions in the feature space. The so-called curse of dimensionality is a well-known phenomenon that appears in many PR applications. It implies that it is advantageous to reduce the number of features in order to reduce computational costs while keeping similar levels of performance, and in some cases even to improve the classification accuracy.

In a PR application, a well-designed feature should have the three following general properties:

- *Invariance to irrelevancies* A good feature should be invariant to irrelevant transformations of the input signal. In an audio context, irrelevancies can include signal quality with respect to noise and bandwidth, the number of channels, or the overall amplitude scaling. To ensure

similar classification rates across a wide range of audio formats or qualities, a classification system should consider such variations of quality irrelevant.
- *Good discriminative power* A feature should take similar values within a given class, but very different values across classes.
- *Uncorrelation to other features* Redundant information should be avoided in the feature space. Each new feature should add as much new information about the object as possible.

Features were selected in a completely systematic way, consisting of two steps. The first step corresponds to the criterium of invariance and consists of the test on robustness to irrelevancies explained in the next subsection. The second step corresponds to the criteria of discriminative power and uncorrelation and its practical implementation is the feature subset selection algorithm described in Section 5.2.

5.1 Tests on Robustness to Irrelevancies

Two tests were carried out to obtain the features that were most sensitive to the addition of noise and to bandwidth changes. To test the robustness against noise, we compared a selection of audio samples from the database with the same examples mixed with white Gaussian noise of −25 dB rms power. To evaluate the bandwidth robustness, the same samples were subjected to low-pass filtering with a cut-off frequency of 11025 Hz. The 20 features that were most susceptible in each of the tests were discarded. Table 1 shows the 10 features that were least susceptible to noise (10 best features) and the 10 most susceptible features (10 worst). Table 2 shows the corresponding results of the filtering test. For a more detailed explanation of the tests, see [19].

The following general conclusions can be drawn from the results of the noise test:

- The DS subfeatures are especially robust to noisy changes in the signal.
- The M and DM subfeatures are highly sensitive to noise in most cases.

Observing the results of the filtering test, we can conclude the following:

- MFCCs are highly robust to low-pass filtering (except for their DM variants).
- The M and DM subfeatures are highly sensitive to low-pass filtering in most cases.
- The predictability ratio feature is extremely sensitive to low-pass filtering in all of its four subfeature variants. The four predictability-related subfeatures belong to the bottom five of the ranking.

As a result of both noise and filtering tests, a total of 32 features (the ones that appear at least once in both bottom 20 ranking lists) were discarded in this first step.

5.2 Feature Subset Selection

The remaining 58 features were subjected to a feaure selection algorithm that selects a subset of features containing the highest class discriminating power. A vectorial sequential feedforward algorithm was used to search for a feature subset that maximized class separability [28].

More formally, given a feature set $\mathscr{X} = \{f_i | i = 1, \ldots, N\}$, the problem consists of finding a subset $\mathscr{Y} = \{f_{i1}, f_{i2}, \ldots, f_{iM}\}$ with $M < N$ that maximizes a given objective function $J(\mathscr{Y})$. The objective function measures the class separability and is given by

$$J = \frac{|\boldsymbol{S}_\mathrm{B}|}{|\boldsymbol{S}_\mathrm{W}|} \tag{11}$$

where $\boldsymbol{S}_\mathrm{B}$ is the between-class scatter matrix,

$$\boldsymbol{S}_\mathrm{B} = \sum_{k=1}^{C} N_k(\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^\mathrm{T} \tag{12}$$

and $\boldsymbol{S}_\mathrm{W}$ is the within-class scatter matrix,

$$\boldsymbol{S}_\mathrm{W} = \sum_{k=1}^{C} \sum_{\boldsymbol{x} \in \omega_k} (\boldsymbol{x} - \boldsymbol{\mu}_k)(\boldsymbol{x} - \boldsymbol{\mu}_k)^\mathrm{T}. \tag{13}$$

Here $\boldsymbol{\mu}_k$ is the mean vector of the N_k samples in class ω_k, $\boldsymbol{\mu}$ is the mean vector of all available N samples, and C is the number of classes.

The sequential feedforward algorithm consists of the following steps (s being the iteration index):

Table 1. Selected results of noise test.

Best Features*	Worst Features†
1. Low energy rate	90. 1st MFCC/DM
2. Beat histogram entropy	89. Audio spectrum centroid/M
3. Root mean square/DS	88. Kurtosis/DM
4. 4th MFCC/DS	87. 5th MFCC/M
5. 5th MFCC/DS	86. Rolloff/DM
6. Root mean square/S	85. Audio spectrum flatness/DM
7. 3rd MFCC/DS	84. Modified harmonic ratio/DM
8. Beat histogram/DS	83. Kurtosis/DS
9. Audio spectrum spread/DS	82. 3rd MFCC/DM
10. Beat histogram/S	81. Kurtosis/S

* Most robust features.
† Features most susceptible to the addition of white noise.

Table 2. Selected results of bandwidth test.

Best Features*	Worst Features†
1. 2nd MFCC/DS	90. Predictability ratio/S
2. 1st MFCC/DS	89. Predictability ratio/M
3. 5th MFCC/DS	88. Predictability ratio/DM
4. 3rd MFCC/DS	87. 5th MFCC/DM
5. 4th MFCC/S	86. Predictability ratio/DS
6. 2nd MFCC/M	85. Kurtosis/DM
7. 4th MFCC/M	84. Loudness/DM
8. 3rd MFCC/M	83. Rolloff/S
9. 5th MFCC/S	82. Audio spectrum flatness/DM
10. Low energy rate	81. Rolloff/DS

* Most robust features.
† Features most susceptible to low-pass filtering.

1) Start with the empty feature set $\mathcal{Y}_0 = \{\emptyset\}$.

2) Out of the features that have not yet been chosen, select the one feature f^+ that maximizes the objective function in combination with the previously selected features,

$$f^+ = \underset{f_i \in \mathcal{X}-\mathcal{Y}_s}{\operatorname{argmax}} \{J(\mathcal{Y}_s \cup f_i)\}.$$

3) Update $\mathcal{Y}_{s+1} = \mathcal{Y}_s \cup f^+, \quad s \to s + 1$.

4) Go to 2).

The algorithm yields a list in which the 58 features are sorted according to their class-separating ability. It also ensures that consecutively selected features are as uncorrelated as possible.

The hierarchical classification scheme has a parallel approach in the context of feature selection. Instead of using the whole training database to obtain a single list of selected features, a genre-dependent feature selection, or hierarchical feature selection, was used, in which only the training samples belonging to the current branch in the classification tree are used to evaluate the separability of the current two, three, or four classes.

As a result this algorithm is run for each split in the tree, with a local number of classes ranging from $C = 2$ to $C = 4$. Thus a set of nine feature lists was obtained, one for each split. Each list shows the features most appropriate for distinguishing between a given set of music or audio subgenres. Tables 3–5 show the three best features for each split. Tables listing the 20 best features at each split can be found in [25].

The following conclusions can be drawn from the results of the genre-dependent feature selection:

- While the M,S, and DS subfeatures appear often among the 20 best features at each split, the DM subfeature appears only four times, that is, the mean of the derivative is not an effective texture-based measure for classification. This can be explained by the fact that the mean of the derivative is very likely to take values close to zero.
- The modified version of the harmonic ratio shows a better overall performance for classification purposes than the original MPEG-7 definition. It appears eight times among the best 20 tables, whereas the original does not appear at all. It performs particularly well in separating chamber music and orchestral subgenres.
- The new rhythmic regularity feature has excellent separating performance in all splits except for the speech and the classical splits. In all other splits it belongs to the top three of the ranking.
- The zero crossings/DS feature is an excellent separator. It tops the list on four occasions, namely, separation between classical and nonclassical music and three classical subseparations.
- Despite its simplicity, the approximation of loudness used works fairly well as a separator, mostly at the highest levels in the hierarchy. This is a motivation for the future usage of more sophisticated perceptual loudness definitions as a feature.
- It is a surprising result that beat strength features play a key role in separating male speech, female speech, and speech with background. However, this phenomenon may be explained in that word and phrase onsets in speech signals produce strong "rhythmic" peaks on the histogram, while noisy or musical background causes these amplitude jumps to be smaller.

6 CLASSIFICATION

A k-nearest neighbor (k-NN) classifier and a Gaussian mixture model (GMM) classifier were evaluated in both their direct and hierarchical variants. The very intuitive

Table 3. Selected results of feature subset selection.*

Speech/Music/Background	Male/Female/Speech + Background	Classical/Nonclassical
1. 2nd MFCC/S	Beat histogram entropy	Zero crossing/DS
2. 4th MFCC/DS	4th MFCC/S	Loudness/M
3. Rhythmic regularity	Loudness/S	Rhythmic regularity

* Three best features for each split (part 1).

Table 4. Selected results of feature subset selection.*

Chamber Music/Orchestral Music	Rock/Pop/Jazz	Chamber Subgenres
1. Zero crossings/DS	Rhytmic regularity	Zero crossings/DS
2. Flux/DS	1st MFCC/M	Flux/M
3. Zero crossings/M	3rd MFCC/M	Audio spectrum flatness/M

* Three best features for each split (part 2).

Table 5. Selected results of feature subset selection.*

Orchestral Subgenres	Hard Rock/Soft Rock	Pop Subgenres
1. Zero crossings/DS	1st MFCC/M	Rhythmic regularity
2. Flux/DS	Rhythmic regularity	1st MFCC/M
3. 2nd MFCC/M	2nd MFCC/M	4th MFCC/S

* Three best features for each split (part 3).

nearest neighbor rule consists of assigning to the unlabeled feature vector the label of the training feature vector that is nearest to it in the feature space. The distance can be measured using one of several existing metrics, but usually the Euclidean distance is used, which is given by

$$d(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sqrt{\sum_{i=1}^{D} (x_{1,i} - x_{2,i})^2} \tag{14}$$

where D is the total number of dimensions (features).

The k-NN rule extends the NN rule by examining the k nearest training samples to the observed vector. It assigns the label which is most frequent among these k samples.

The GMM classifier models each class as a linear combination of Gaussian or normal densities, that is, each class k is represented by the multidimensional conditional density

$$p(\boldsymbol{x}|\omega_k) = \sum_{m=1}^{M} w_{km}\, p_{km}(\boldsymbol{x}) \tag{15}$$

where ω_k is the event "belongs to class k", $\boldsymbol{x}$ denotes a feature vector, w_{km} are the weights of the mixture, M is the total number of densities in the mixture (called components), and p_{km} is a normal density. The conditional density $p(\boldsymbol{x}|\omega_k)$ is also called the likelihood of class k with respect to $\boldsymbol{x}$. In the particular case $M = 1$, each class is modeled by a normal distribution, and the classifier reduces to a simple Gaussian (GS) classifier.

In the training phase the values of w_{km}, the mean vectors, and the covariance matrices for each component in each class, which are called the parameters of that class, are estimated using an algorithm called expectation maximization [29]. Unlike the k-NN classifier, which needs to store all the training feature vectors in order to compute the distances to the input feature vector, the GMM classifier only needs to store the set of estimated parameters for each class. When an input vector is to be classified, its conditional density in each of the classes $p(\boldsymbol{x}|\omega_k)$ is computed using the estimated parameters. The class for which the density value is highest is the class chosen for that vector. This decision rule is called the maximum-likelihood criterion [29], and the condition for applying it is that the different classes be equally probable a priori, which is usually assumed in an audio classification system.

Experiments showed that both classifiers achieved very similar classification accuracy in the direct as well as the hierarchical versions. However, the GMM classifier reached its maximum level of classification accuracy for about 20 features, compared with close to 40 features needed by the k-NN to achieve similar rates (see Fig. 5). This, together with the fact that it reduces the training set to a set of class parameters, makes the GMM much more computationally efficient. Therefore it was chosen for the final implementation of our system.

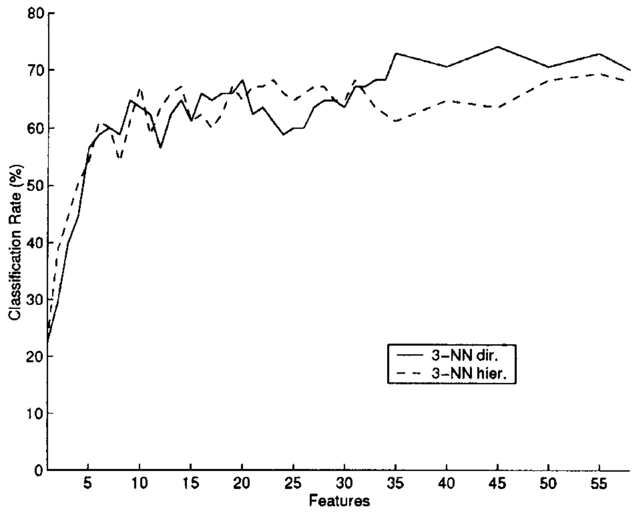
A three-component GMM was chosen because it showed a slightly better performance than 1-, 2-, 4-, 5- or 6-component GMMs. At approximately 20 features the performance ceased to increase, and in some evaluation experiments it even decreased. For these reasons the number of features was fixed at 20, that is, at each classification step the system extracts the 20 best features from the signal, as indicated by the feature selection list corresponding to the current tree split. It should be noted that the optimal number of features depends closely on the number of training samples.

6.1 Hierarchical Classification

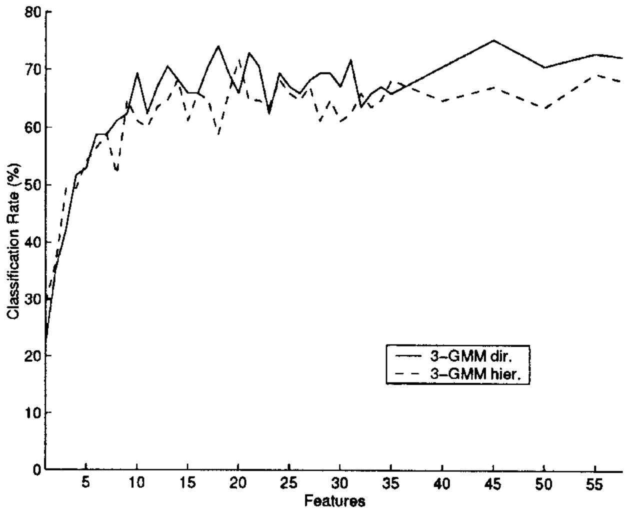
As mentioned before, a special effort was made in exploring a hierarchical classification approach which consisted of a tree-based succession of class decisions corresponding to the class taxonomy depicted in Fig. 1. In this way level 1 of the taxonomy corresponds to a three-class classification problem. Following the decision of this first problem, we then switch to one of the three- or two-class problems at the second level, and so on. This contrasts with the more common direct approach, which performs a single decision out of all 17 classes.

An incentive for using the hierarchical method was the advantages it offers in comparison to the direct approach [30]:

- A hierarchical approach allows to account for the class dependency of the features. This is exactly the motiva-



(a)



(b)

Fig. 5. Selected results for all-class classification experiments. (a) 3-NN classifier. (b) 3-GMM classifier. — direct; --- Hierarchical.

tion for the genre-dependent feature selection addressed in the previous section.

- It allows the errors to be more acceptable than in the case of a direct classification. For example, if a symphonic music sample is wrongly classified as orchestral music with soloist, it is not so bad as if it were classified as hip-hop. Dividing the decision into subdecisions makes the errors concentrate within a given subgenre.
- It closely reflects the underlying audio taxonomy, thus allowing to evaluate the separability of broad commonly used genres, such as pop, rock, and jazz, and their suitability for automatic classification purposes.
- It provides the framework for the future design of more sophisticated genre-dependent features.
- It makes future expansions of the taxonomy easier. If a new class were added to the direct scheme, the feature selection algorithm would have to be run again with all training samples. Furthermore, each class would have to be retrained according to the new result of the feature selection. In contrast, in a hierarchical approach only the genre branch to which a new class is added should be modified with respect to feature selection and training, the rest of the models remaining unchanged. In this way end users would be able to create their own subsplits on the taxonomy without needing the whole training database for the rest of the genres.

However, the hierarchical method has the following three major drawbacks:

- The classification rates are multiplicative across levels. For example, a signal classified correctly as string quartet must have been correctly classified as chamber music at the previous level, as classical music at the previous level and as music at the highest level. As it has been found by the experiments explained in the next section, this leads to a slightly poorer performance than the direct case. To compensate for the higher possibility of error, the genre-dependent features must be very well designed to fit their particular classes.
- It presents more complexity in the implementation.
- It is more computationally expensive, not only because more classification decisions have to be made for a given input signal, but also because it is likely that more features have to be computed than in the direct case.

7 EVALUATION

All the implementations and computations that have been detailed in the previous sections have been simulated and tested in a scripting language, which is too slow for a practical application. Therefore the selected features and the classifier algorithm (3-GMM) were subsequently implemented as a prototype classification application in the C++ language. This application implements the audio preprocessing, the feature extraction, and the classification. This section contains the results of the evaluation of this final application regarding classification accuracy and computational performance. Furthermore, the evaluation experiments of the previous section were simple and based on the all-class classification rate. In the following, the C++ application was subjected to more detailed and demanding evaluation procedures.

To obtain a realistic estimation of performance, the system was evaluated using the so-called K-fold cross-validation method, in which the evaluation is iterated K times, each time using a randomly selected test set consisting of $100/K\%$ of the samples of the whole database. Then the performances across the iterations are averaged to obtain the final estimate of the classification rate R.

In addition it was imposed that each class be represented by the same number of samples in the test set, resulting in a stratified crossvalidation. Also, each sample in the database belongs to only one of the test sets, so that the union of all test sets corresponds to the whole database. In other words, it is assured that each example is used exactly once for testing, the other $K - 1$ times for training. A value of $K = 10$ is a common choice in the evaluation of pattern recognizers and has also been adopted in this work.

7.1 Evaluation in Single-Vector Mode

We first evaluated the classification accuracy of the program operating in single-vector mode. As explained in Section 4, in this mode a single feature vector is computed for each file. Table 6 shows the percentages of correct classifications (mean plus standard deviation across the cross-validation experiments) for the single-vector hierarchical approach and for each of the tree splits, as well as an all-class classification rate, which takes into account all 17 classes. For each split on the tree two different performance indications will be given:

- *Cumulative performance* Percentage of samples of the test set correctly classified.
- *Independent performance* Percentage of samples correctly classified at level i that have been correctly classified at level $i + 1$.

The cumulative performance is a more demanding measure, since it considers the samples that have been incorrectly classified elsewhere in the tree, accounting for the multiplicative nature of the performance of tree-based classifiers. On the other hand, the independent perfor-

Table 6. Classification performance rates using hierarchical approach in single-vector mode (means plus standard deviations across experiments of cross validation).

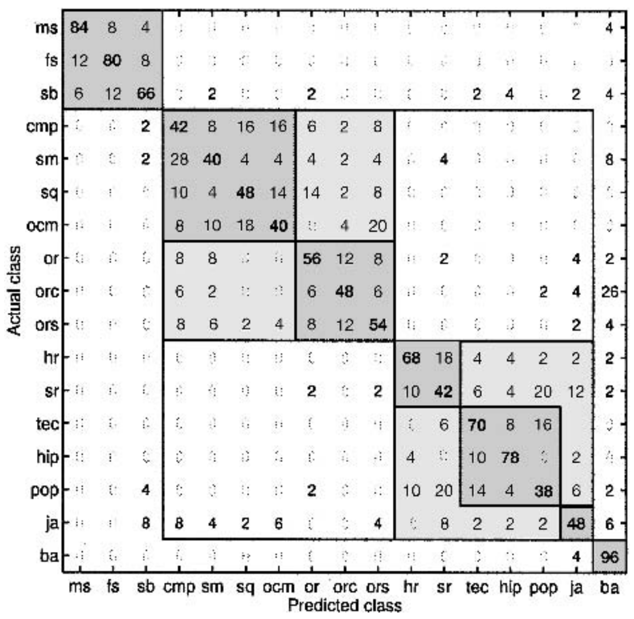
Split	Cumulative Performance	Independent Performance
Speech/music background	94.59 ± 1.77	94.59 ± 1.77
Male/female/speech + background	76.67 ± 8.46	82.31 ± 8.63
Classical/Nonclassical	91.08 ± 3.68	96.08 ± 2.02
Chamber music/orchestral music	74.29 ± 7.25	81.52 ± 7.88
Rock/pop/jazz	63.67 ± 6.17	70.33 ± 8.65
Chamber subgenres	42.50 ± 12.08	54.67 ± 13.92
Orchestral subgenres	52.67 ± 10.63	75.21 ± 11.83
Hard rock/soft rock	55.00 ± 16.50	79.52 ± 20.18
Pop subgenres	62.00 ± 9.96	76.15 ± 9.55
All classes	58.71 ± 2.85	

mance is useful for estimating the capability of the system in separating a given set of subclasses. For the particular case of the speech/music/background split, cumulative and independent performances will be identical. The all-class performance is implicitly cumulative.

A more detailed information about the classification performance is given by the so-called confusion matrix (Fig. 6). Its rows correspond to the actual classes and its columns to the predicted classes. For example, an entry of 8 in element (1, 2) of the matrix denotes that 8% of the male speech samples were wrongly classified as female speech. Correct classifications correspond to the diagonal. The square margins and the different grades of shading denote the splits on the hierarchy. The class abbreviations were introduced in Fig. 1.

The following conclusions can be drawn from the observation of the confusion matrix:

- It is seen clearly that the classification errors tend to appear within the same class groups or subgroups, which indicates the ability of the hierarchical approach to produce acceptable errors.
- Many music samples were classified wrongly as background. Perhaps the most surprising result of the confusion matrix is the high number of choral music samples (26%) that have been classified wrongly as background noise. On the other hand, only 4% of the background samples were classified as music.
- All the nonclassical music samples misclassified as chamber music are jazz examples. This is musicologically consistent, since with regard to instrumentation, most of the jazz examples contained in the database are chamber-music-like (trios, quartets, and so on).
- The matrix also shows the difficulty of separating the rock and pop classes. In particular the most difficult genres in the hierarchy are soft rock (42% of correct classifications) and pop (38% correct). It can also be seen that many soft rock examples were classified as pop and vice versa (in both cases 20%). This corresponds to the inherently fuzzy nature of these particular genres, which are understood in many different ways across individuals. A study by Soltau [31] has confirmed experimentally the difficulty of human differentiation between rock and pop.
- The musical genres with the best classification rates are rap/hip-hop (78%) and techno/dance (70%). This indicates that these classes are very distinct as their specific examples have similar characteristics within their class (low within scatter) but dissimilar characteristics when compared with the rest of the classes (high between scatter).



Actual class \ Predicted class	ms	fs	sb	cmp	sm	sq	ocm	or	orc	ors	hr	sr	tec	hip	pop	ja	ba
ms	84	8	4														4
fs	12	80	8														
sb	6	12	66		2			2					2	4		2	4
cmp			2	42	8	16	16	6	2	8							
sm			2	28	40	4	4	4	2	4		4					8
sq				10	4	48	14	14	2	8							
ocm				8	10	18	40		4	20							
or				8	8			56	12	8		2				4	2
orc				6	2			6	48	6					2	4	26
ors				8	6	2	4	8	12	54						2	4
hr											68	18	4	4	2	2	2
sr								2		2	10	42	6	4	20	12	2
tec												6	70	8	16		
hip											4		10	78		2	
pop			4					2			10	20	14	4	38	6	2
ja			8	8	4	2	6			4		8	2	2	2	48	6
ba																4	96

Fig. 6. Confusion matrix for hierarchical approach in single-vector mode. An entry of *R* in element (*i*, *j*) means that *R*% of test samples of class *i* were classified as belonging to class *j*. Correct classification rates correspond to the diagonal. For class codes, see Fig. 1.

The evaluation was repeated for the direct approach. The results are shown on Table 7. It should be noted that in this case a single feature list was used which was obtained by running the feature-selection algorithm over the entire 17-class training database.

7.2 Evaluation in Texture-Window Mode

To evaluate the system operating in the texture-window mode, a 10-fold cross-validation experiment was carried out for each of the following seven texture-window sizes: 0.5, 1, 2, 5, 10, 15, and 20 seconds, using the hierarchical approach. It should always be kept in mind that all the training and test samples are approximately 30 seconds long.

Fig. 7 shows the classification rate for different texture-window sizes. The curves indicate the levels of independent performance averaged at each of the four levels on the tree. When compared with the results of the single-vector mode (Table 8), it can be seen that with the largest texture-window sizes, the levels of performance reached by the texture-window mode are similar to those of the single-vector mode.

It can also be seen that the curves cease to increase at a window size of 15 seconds. This indicates that observing longer blocks from the files does not provide much additional information, assuming the audio content is reasonably homogeneous. Under this condition, the system is

Table 7. Classification performance rates using direct approach in single-vector mode (means plus standard deviations across experiments of cross validation).

Split	Cumulative Performance	Independent Performance
Speech/music/background	96.35 ± 1.70	96.35 ± 1.70
Male/female/speech + background	76.67 ± 9.03	81.00 ± 9.19
Classical/nonclassical	94.31 ± 3.48	96.67 ± 2.45
Chamber music/orchestral music	75.43 ± 7.15	78.06 ± 6.81
Rock/pop/jazz	65.33 ± 5.49	71.83 ± 9.38
Chamber subgenres	50.50 ± 9.26	63.05 ± 13.24
Orchestral subgenres	52.00 ± 15.65	75.86 ± 18.26
Hard rock/soft rock	59.00 ± 19.69	78.91 ± 21.09
Pop subgenres	58.67 ± 16.87	71.57 ± 16.04
All classes	59.76 ± 5.23	

expected to maintain similar performance with longer texture windows.

7.3 Evaluation of Computational Performance

The final application implemented on a Pentium III processor at 1000 MHz proved to perform feature extraction about one-third faster than in real time. As noted in Section 1, real-time feature extraction should not be confused with real-time classification.

8 CONCLUSIONS

As can be seen, the classification rates are very similar for both direct and hierarchical approaches. However, the hierarchical approach features the additional advantages mentioned in Section 6.1, and thus it was selected for the final implementation of the system.

The classification rates differ substantially across levels of the tree, showing the varying grades of difficulty in separating each subset of audio classes. The best independent performances were achieved at the highest levels in the tree. For example 94.59% accuracy was achieved in differentiating between speech, background, and music, 96.08% in separating classical from nonclassical music, and 81.52% in separating chamber music from orchestral music.

In contrast, the main difficulties arise in the most specific genres at the lowest levels of the tree, especially in the case of the four chamber music subgenres, where the total classification accuracy was 54.67%. The lower levels of the tree, as well as the high number of classes considered, make the all-class classification rate drop to 58.71%.

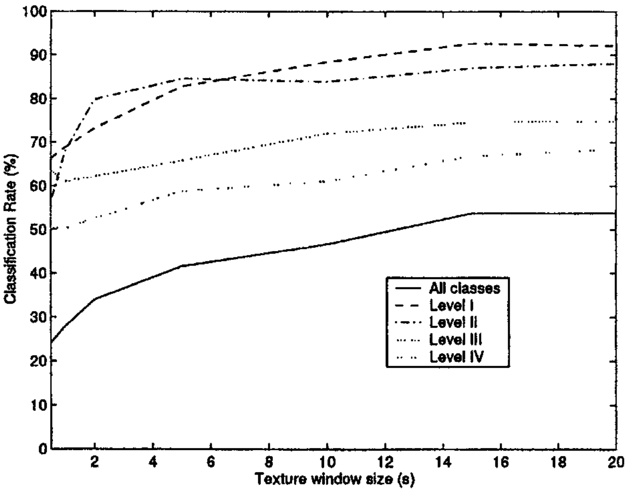


Fig. 7. Results of texture-window-mode evaluation.

Table 8. Independent performances averaged across levels for hierarchical approach in single-vector mode.

Level	Single Vector
I	94.59
II	89.20
III	75.92
IV	71.39
All classes	58.71

To achieve higher rates at these levels, more elaborate genre-specific features are needed.

It is not possible to make a reliable comparison across systems with respect to classification accuracy, most of all because of the different numbers of classes considered. One possibility to benchmark a classification system is to compare its accuracy with the one that would result if the classes were chosen randomly. In a system with C classes the random accuracy would be $100/C\%$. The system that is most similar to the one described in this work with respect to taxonomy complexity and classification method is the one by Tzanetakis and Cook [16]. They reported an independent performance of 61% when classifying into 10 classes (10% random), compared with 59.76% (direct approach) and 58.71% (hierarchical approach) achieved by the authors when classifying into 17 classes (5.88% random).

The three-class highest classification level (speech/music/background) for which accuracies of 96.35% (direct approach) and 94.59% (hierarchical approach) were obtained can be compared with the systems of Zhang and Kuo [10] and Lu and Jiang [17], which achieved performances of 90% and 96.51%, respectively, with exactly the same three classes. The differentiation between male speech, female speech, and speech with background (81% direct, 82.31% hierarchical) can be compared to the equivalent classification in Tzanetakis and Cook [16] (74%).

The implemented prototype application can constitute the basis of a practical file-based classifier operating at the highest levels of the tree. However, the 17-class performance is still too low for considering a practical application comprising all classes in the taxonomy. To improve this overall performance, an obvious direction for future research is the design of sophisticated genre-dependent features. Possibilities include, for example, a measure of overall distortion for detecting hard rock, singer detection to distinguish opera from other classical genres, a measurement of reverberation to discern between chamber music and symphonic music, more sophisticated loudness models, or the analysis of higher level musical characteristics such as harmony or melody.

Other possible directions include the evaluation of other classification algorithms (such as support vector machines or hidden Markov models), the exploration of other dimensionality reduction methods (such as linear discriminant analysis or other appropriate transformations of the feature space), and the implementation of real-time classification.

9 REFERENCES

[1] J. Foote, "An Overview of Audio Information Retrieval," *Multimedia Sys.,* vol. 7, no. 1 (1999).

[2] J. S. Downie, "Music Information Retrieval," in *Annual Review of Information Science and Technology,* vol. 37, B. Cronin, Ed. (Information Today, Medford, NJ, 2003), pp. 295–340.

[3] B. Feiten and S. Günzel, "Automatic Indexing of Sound Database Using Self-Organizing Neural Nets," *Computer Music J.,* vol. 18, pp. 53–65 (1994 Mar.).

[4] A. Rauber and M. Früwirth, "Automatically Ana-

lyzing and Organizing Music Archives," in *Proc. 5th Eur. Conf. on Research and Advanced Technology for Digital Libraries (ECDL)* (2001 Sept.).

[5] M. Frühwirth and A. Rauber, "Self-Organizing Maps for Content-Based Music Clustering," in *Proc. 12th Italian Workshop on Neural Nets (WIRN01)* (2001 May).

[6] E. Pampalk, "Islands of Music: Analysis, Organization, and Visualization of Music Archives," M.S. thesis, Technische Universität Wien, Vienna, Austria (2001 Dec.).

[7] E. Wold, T. Blum, D. Keislar, and J. Wheaton, "Content-Based Classification, Search and Retrieval of Audio," *IEEE Multimedia,* pp. 27–36 (1996 Fall).

[8] E. Scheirer and M. Slaney, "Construction and Evaluation of a Robust Multifeature Speech/Music Discriminator," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)* (1997).

[9] J. Foote, "A Similarity Measure for Automatic Audio Classification," in *Proc. AAAI 1997 Spring Symp. on Intelligent Integration and Use of Text, Image, Video, and Audio Corpora* (Stanford, CA, 1997).

[10] T. Zhang and J. Kuo, "Hierarchical System for Content-Based Audio Classification and Retrieval," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)* (1998).

[11] T. Zhang and J. Kuo, "Content-Based Classification and Retrieval of Audio," in *Proc. Conf. on Advanced Signal Processing Algorithms, Architectures and Implementations,* vol. VIII (1998 July).

[12] D. Pye, "Content-Based Methods for the Management of Digital Music," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)* (2000).

[13] M. Casey, "General Sound Classification and Similarity in MPEG-7," *Organized Sound,* vol. 6, no. 2 (2002).

[14] B. S. Manjunath, P. Salembier, and T. Sikora, Eds., *Introduction to MPEG-7* (Wiley, New York, 2002)

[15] G. Tzanetakis, G. Essl, and P. Cook, "Automatic Musical Genre Classification of Audio Signals," in *Proc. 2nd Ann. Int. Symp. on Music Information Retrieval (ISMIR)* (2001).

[16] G. Tzanetakis and P. Cook, "Musical Genre Classification of Audio Signals," *IEEE Trans. Speech and Audio Process.,* vol. 10, pp. 293–302 (2002 July).

[17] L. Lu and H. Jiang, "Content Analysis for Audio Classification and Segmentation," *IEEE Trans. Speech and Audio Process.,* vol. 10, pp. 504–516 (2002 Oct.).

[18] A. T. Lindsay and J. Herre, "MPEG-7 and MPEG-7 Audio—An Overview," *J. Audio Eng. Soc.,* vol. 49, pp. 589–594 (2001 July/Aug.).

[19] J. J. Burred and A. Lerch, "A Hierarchical Approach to Automatic Musical Genre Classification," in *Proc. 6th Int. Conf. on Digital Audio Effects (DAFX)* (London, UK, 2003).

[20] G. Peeters and X. Rodet, "Hierarchical Gaussian Tree with Inertia Ratio Maximization for the Classification of Large Musical Instrument Databases," in *Proc. 6th Int. Conf. on Digital Audio Effects (DAFX)* (London, UK, 2003).

[21] T. Lambrou, P. Kudumakis, R. Speller, M. Sandler, and A. Linney, "Classification of Audio Signals Using Statistical Features on Time and Wavelet Transform Domains," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)* (1998).

[22] B. Logan, "Mel Frequency Cepstral Coefficients for Music Modeling," in *Proc. 1st Ann. Int. Symp. on Music Information Retrieval (ISMIR)* (2000).

[23] ISO/IEC FDIS 15938-4, "Information Technology—Multimedia Content Description Interface—Part 4: Audio," International Standards Organization, Geneva, Switzerland (2001).

[24] E. D. Scheirer, "Tempo and Beat Analysis of Acoustic Musical Signals," *J. Acoust. Soc. Am.,* vol. 103, pp. 588–601 (1998 Jan.).

[25] J. J. Burred, "An Objective Approach to Content-Based Audio Signal Classification," M.S. thesis, Technische Universität Berlin, Berlin, Germany (2003 May).

[26] E. Zwicker and H. Fastl, *Psychoacoustics* (Springer, Berlin, 1999).

[27] B. C. J. Moore, B. R. Glasberg, and T. Baer, "A Model for The Prediction of Thresholds, Loudness, and Partial Loudness," *J. Audio Eng. Soc.,* vol. 45, pp. 224–240 (1997 Apr.).

[28] K. Fukunaga, *Introduction to Statistical Pattern Recognition* (Academic Press, New York, 1990).

[29] R. Duda, P. Hart, and D. Stork, *Pattern Classification,* (Wiley, New York, 2000).

[30] J. J. Aucouturier and F. Pachet, "Representing Musical Genre: A State of the Art," *J. New Music Research,* vol. 32, no. 1 (2003).

[31] H. Soltau, "Erkennung von Musikstilen" (in German), M.S. thesis, Institut für Logik, Komplexität und Deduktionssysteme, Universität Karlsruhe, Karlsruhe, Germany (1997 May).

THE AUTHORS

J. J. Burred

A. Lerch

Juan José Burred was born in Valencia, Spain, in 1978. He received a Telecommunication Engineering degree from the Polytechnic University of Madrid in 2004. He wrote his master's thesis about audio signal classification at the Berlin-based company zplane.development, in cooperation with the Technical University of Berlin.

In 2003 Mr. Burred joined the Communication Systems Group of the Technical University Berlin as a student research assistant within the MPEG-7 Audio project. Since 2004 he has been a Ph.D. student and research assistant at the same institute. His research interests include audio content analysis (especially music content analysis), audio source separation, and auditory scene analysis. He also has degrees in piano and in music theory from the Madrid Professional Music Conservatory, and he has been active as a musician.

Mr. Burred was awarded the ETV Prize 2004 from the Berlin Association of Electrical Engineers for the best master's thesis in electrical engineering.

●

Alexander Lerch studied tonmeister at the University of Arts Berlin, in Germany, and electrical engineering with focus on digital audio signal processing and acoustics at the Technical University Berlin. He completed his studies (summa cum laude) at the Technical University.

In 2001 Mr. Lerch cofounded zplane.development, a company dedicated to the research and development of new audio technologies. He was a visiting lecturer at the University of Arts Berlin. He is member of the Audio Engineering Society, the Deutsche Gesellschaft für Akustik, the Institute of Electrical and Electronics Engineers, and the Verband deutscher Tonmeister.