Musical Genre Classification of Audio Signals

Article *in* IEEE Transactions on Speech and Audio Processing · August 2002

DOI: 10.1109/TSA.2002.800560 · Source: IEEE Xplore

2 authors:

George Tzanetakis
University of Victoria
231 PUBLICATIONS **8,061** CITATIONS
SEE PROFILE

Perry R. Cook
Princeton University
228 PUBLICATIONS **9,851** CITATIONS
SEE PROFILE

Musical Genre Classification of Audio Signals

George Tzanetakis, *Student Member, IEEE,* and Perry Cook, *Member, IEEE*

***Abstract*—Musical genres are categorical labels created by humans to characterize pieces of music. A musical genre is characterized by the common characteristics shared by its members. These characteristics typically are related to the instrumentation, rhythmic structure, and harmonic content of the music. Genre hierarchies are commonly used to structure the large collections of music available on the Web. Currently musical genre annotation is performed manually. Automatic musical genre classification can assist or replace the human user in this process and would be a valuable addition to music information retrieval systems. In addition, automatic musical genre classification provides a framework for developing and evaluating features for any type of content-based analysis of musical signals.**

In this paper, the automatic classification of audio signals into an hierarchy of musical genres is explored. More specifically, three feature sets for representing timbral texture, rhythmic content and pitch content are proposed. The performance and relative importance of the proposed features is investigated by training statistical pattern recognition classifiers using real-world audio collections. Both whole file and real-time frame-based classification schemes are described. Using the proposed feature sets, classification of 61% for ten musical genres is achieved. This result is comparable to results reported for human musical genre classification.

***Index Terms*—Audio classification, beat analysis, feature extraction, musical genre classification, wavelets.**

I. Introduction

MUSICAL genres are labels created and used by humans for categorizing and describing the vast universe of music. Musical genres have no strict definitions and boundaries as they arise through a complex interaction between the public, marketing, historical, and cultural factors. This observation has led some researchers to suggest the definition of a new genre classification scheme purely for the purposes of music information retrieval [1]. However even with current musical genres, it is clear that the members of a particular genre share certain characteristics typically related to the instrumentation, rhythmic structure, and pitch content of the music.

Automatically extracting music information is gaining importance as a way to structure and organize the increasingly large numbers of music files available digitally on the Web. It is very likely that in the near future all recorded music in human history will be available on the Web. Automatic music analysis will be one of the services that music content distribution vendors will use to attract customers. Another indication of the increasing importance of digital music distribution is the legal attention that companies like Napster have recently received.

Genre hierarchies, typically created manually by human experts, are currently one of the ways used to structure music content on the Web. Automatic musical genre classification can potentially automate this process and provide an important component for a complete music information retrieval system for audio signals. In addition it provides a framework for developing and evaluating features for describing musical content. Such features can be used for similarity retrieval, classification, segmentation, and audio thumbnailing and form the foundation of most proposed audio analysis techniques for music.

In this paper, the problem of automatically classifying audio signals into an hierarchy of musical genres is addressed. More specifically, three sets of features for representing timbral texture, rhythmic content and pitch content are proposed. Although there has been significant work in the development of features for speech recognition and music–speech discrimination there has been relatively little work in the development of features specifically designed for music signals. Although the timbral texture feature set is based on features used for speech and general sound classification, the other two feature sets (rhythmic and pitch content) are new and specifically designed to represent aspects of musical content (rhythm and harmony). The performance and relative importance of the proposed feature sets is evaluated by training statistical pattern recognition classifiers using audio collections collected from compact disks, radio, and the Web. Audio signals can be classified into an hierarchy of music genres, augmented with speech categories. The speech categories are useful for radio and television broadcasts. Both whole-file classification and real-time frame classification schemes are proposed.

The paper is structured as follows. A review of related work is provided in Section II. Feature extraction and the three specific feature sets for describing timbral texture, rhythmic structure, and pitch content of musical signals are described in Section III. Section IV deals with the automatic classification and evaluation of the proposed features and Section V with conclusions and future directions.

II. Related Work

The basis of any type of automatic audio analysis system is the extraction of feature vectors. A large number of different feature sets, mainly originating from the area of speech recognition, have been proposed to represent audio signals. Typically

Manuscript received November 28, 2001; revised April 11, 2002. This work was supported by the NSF under Grant 9984087, the State of New Jersey Commission on Science and Technology under Grant 01-2042-007-22, Intel, and the Arial Foundation. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. C.-C. Jay Kuo.

G. Tzanetakis is with the Computer Science Department, Princeton University, Princeton, NJ 08544 USA (e-mail: gtzan@cs.princeton.edu).

P. Cook is with the Computer Science and Music Departments, Princeton University, Princeton, NJ 08544 USA (e-mail: prc@cs.princeton.edu).

Publisher Item Identifier 10.1109/TSA.2002.800560.

they are based on some form of time-frequency representation. Although a complete overview of audio feature extraction is beyond the scope of this paper, some relevant representative audio feature extraction references are provided.

Automatic classification of audio has also a long history originating from speech recognition. Mel-frequency cepstral coefficients (MFCC) [2], are a set of perceptually motivated features that have been widely used in speech recognition. They provide a compact representation of the spectral envelope, such that most of the signal energy is concentrated in the first coefficients.

More recently, audio classification techniques that include nonspeech signals have been proposed. Most of these systems target the classification of broadcast news and video in broad categories like music, speech, and environmental sounds. The problem of discrimination between music and speech has received considerable attention from the early work of Saunders [3] where simple thresholding of the average zero-crossing rate and energy features is used, to the work of Scheirer and Slaney [4] where multiple features and statistical pattern recognition classifiers are carefully evaluated. In [5], audio signals are segmented and classified into "music," "speech," "laughter," and nonspeech sounds using cepstral coefficients and a hidden Markov model (HMM). A heuristic rule-based system for the segmentation and classification of audio signals from movies or TV programs based on the time-varying properties of simple features is proposed in [6]. Signals are classified into two broad groups of music and nonmusic which are further subdivided into (music) harmonic environmental sound, pure music, song, speech with music, environmental sound with music, and (non-music) pure speech and nonharmonic environmental sound. Berenzweig and Ellis [7] deal with the more difficult problem of locating singing voice segments in musical signals. In their system, the phoneme activation output of an automatic speech recognition system is used as the feature vector for classifying singing segments.

Another type of nonspeech audio classification system involves isolated musical instrument sounds and sound effects. In the pioneering work of Wold *et al.* [8] automatic retrieval, classification and clustering of musical instruments, sound effects, and environmental sounds using automatically extracted features is explored. The features used in their system are statistics (mean, variance, autocorrelation) over the whole sound file of short time features such as pitch, amplitude, brightness, and bandwidth. Using the same dataset various other retrieval and classification approaches have been proposed. Foote [9] proposes the use of MFCC coefficients to construct a learning tree vector quantizer. Histograms of the relative frequencies of feature vectors in each quantization bin are subsequently used for retrieval. The same dataset is also used in [10] to evaluate a feature extraction and indexing scheme based on statistics of the discrete wavelet transform (DWT) coefficients. Li [11] used the same dataset to compare various classification methods and feature sets and proposed the use of the nearest feature line pattern classification method.

In the previously cited systems, the proposed acoustic features do not directly attempt to model musical signals and therefore are not adequate for automatic musical genre classification. For example, no information regarding the rhythmic structure of the music is utilized. Research in the areas of automatic beat detection and multiple pitch analysis can provide ideas for the development of novel features specifically targeted to the analysis of music signals.

Scheirer [12] describes a real-time beat tracking system for audio signals with music. In this system, a filterbank is coupled with a network of comb filters that track the signal periodicities to provide an estimate of the main beat and its strength. A real-time beat tracking system based on a multiple agent architecture that tracks several beat hypotheses in parallel is described in [13]. More recently, computationally simpler methods based on onset detection at specific frequencies have been proposed in [14] and [15]. The beat spectrum, described in [16], is a more global representation of rhythm than just the main beat and its strength.

To the best of our knowledge, there has been little research in feature extraction and classification with the explicit goal of classifying musical genre. Reference [17] contains some early work and preliminary results in automatic musical genre classification.

III. Feature Extraction

Feature extraction is the process of computing a compact numerical representation that can be used to characterize a segment of audio. The design of descriptive features for a specific application is the main challenge in building pattern recognition systems. Once the features are extracted standard machine learning techniques which are independent of the specific application area can be used.

A. Timbral Texture Features

The features used to represent timbral texture are based on standard features proposed for music-speech discrimination [4] and speech recognition [2]. The calculated features are based on the short time Fourier transform (STFT) and are calculated for every short-time frame of sound. More details regarding the STFT algorithm and the Mel-frequency cepstral coefficients (MFCC) can be found in [18]. The use of MFCCs to separate music and speech has been explored in [19]. The following specific features are used to represent timbral texture in our system.

1) Spectral Centroid: The spectral centroid is defined as the center of gravity of the magnitude spectrum of the STFT

$$C_t = \frac{\sum_{n=1}^{N} M_t[n] * n}{\sum_{n=1}^{N} M_t[n]} \tag{1}$$

where $M_t[n]$ is the magnitude of the Fourier transform at frame t and frequency bin n. The centroid is a measure of spectral shape and higher centroid values correspond to "brighter" textures with more high frequencies.

2) Spectral Rolloff: The spectral rolloff is defined as the frequency R_t below which 85% of the magnitude distribution is concentrated

$$\sum_{n=1}^{R_t} M_t[n] = 0.85 * \sum_{n=1}^{N} M_t[n]. \tag{2}$$

The rolloff is another measure of spectral shape.

3) Spectral Flux: The spectral flux is defined as the squared difference between the normalized magnitudes of successive spectral distributions

$$F_t = \sum_{n=1}^{N} (N_t[n] - N_{t-1}[n])^2 \quad (3)$$

where $N_t[n]$ and $N_{t-1}[n]$ are the normalized magnitude of the Fourier transform at the current frame t, and the previous frame $t-1$, respectively. The spectral flux is a measure of the amount of local spectral change.

4) Time Domain Zero Crossings:

$$Z_t = \frac{1}{2} \sum_{n=1}^{N} |sign(x[n]) - sign(x[n-1])| \quad (4)$$

where the $sign$ function is 1 for positive arguments and 0 for negative arguments and $x[n]$ is the time domain signal for frame t. Time domain zero crossings provide a measure of the noisiness of the signal.

5) Mel-Frequency Cepstral Coefficients: Mel-frequency cepstral coefficients (MFCC) are perceptually motivated features that are also based on the STFT. After taking the log-amplitude of the magnitude spectrum, the FFT bins are grouped and smoothed according to the perceptually motivated Mel-frequency scaling. Finally, in order to decorrelate the resulting feature vectors a discrete cosine transform is performed. Although typically 13 coefficients are used for speech representation, we have found that the first five coefficients provide the best genre classification performance.

6) Analysis and Texture Window: In short-time audio analysis, the signal is broken into small, possibly overlapping, segments in time and each segment is processed separately. These segments are called *analysis windows* and have to be small enough so that the frequency characteristics of the magnitude spectrum are relatively stable (i.e., assume that the signal for that short amount of time is stationary). However, the sensation of a sound "texture" arises as the result of multiple short-time spectrums with different characteristics following some pattern in time. For example, speech contains vowel and consonant sections which have very different spectral characteristics.

Therefore, in order to capture the long term nature of sound "texture," the actual features computed in our system are the running means and variances of the extracted features described in the previous section over a number of analysis windows. The term *texture window* is used in this paper to describe this larger window and ideally should correspond to the minimum time amount of sound that is necessary to identify a particular sound or music "texture." Essentially, rather than using the feature values directly, the parameters of a running multidimensional Gaussian distribution are estimated. More specifically, these parameters (means, variances) are calculated based on the *texture window* which consists of the current feature vector in addition to a specific number of feature vectors from the past. Another way to think of the *texture window* is as a memory of the past. For efficient implementation a circular buffer holding previous feature vectors can be used. In our system, an *analysis window* of 23 ms (512 samples at 22 050 Hz sampling rate) and a *texture window* of 1 s (43 analysis windows) is used.

7) Low-Energy Feature: Low energy is the only feature that is based on the *texture window* rather than the *analysis window*. It is defined as the percentage of *analysis windows* that have less RMS energy than the average RMS energy across the *texture window*. As an example, vocal music with silences will have large low-energy value while continuous strings will have small low-energy value.

B. Timbral Texture Feature Vector

To summarize, the feature vector for describing timbral texture consists of the following features: means and variances of spectral centroid, rolloff, flux, zerocrossings over the texture window (8), low energy (1), and means and variances of the first five MFCC coefficients over the texture window (excluding the coefficient corresponding to the DC component) resulting in a 19-dimensional feature vector.

C. Rhythmic Content Features

Most automatic beat detection systems provide a running estimate of the main beat and an estimate of its strength. In addition to these features in order to characterize musical genres more information about the rhythmic content of a piece can be utilized. The regularity of the rhythm, the relation of the main beat to the subbeats, and the relative strength of subbeats to the main beat are some examples of characteristics we would like to represent through feature vectors.

One of the common automatic beat detector structures consists of a filterbank decomposition, followed by an envelope extraction step and finally a periodicity detection algorithm which is used to detect the lag at which the signal's envelope is most similar to itself. The process of automatic beat detection resembles pitch detection with larger periods (approximately 0.5 s to 1.5 s for beat compared to 2 ms to 50 ms for pitch).

The calculation of features for representing the rhythmic structure of music is based on the wavelet transform (WT) which is a technique for analyzing signals that was developed as an alternative to the STFT to overcome its resolution problems. More specifically, unlike the STFT which provides uniform time resolution for all frequencies, the WT provides high time resolution and low-frequency resolution for high frequencies, and low time and high-frequency resolution for low frequencies. The discrete wavelet transform (DWT) is a special case of the WT that provides a compact representation of the signal in time and frequency that can be computed efficiently using a fast, pyramidal algorithm related to multirate filterbanks. More information about the WT and DWT can be found in [20]. For the purposes of this work, the DWT can be viewed as a computationally efficient way to calculate an octave decomposition of the signal in frequency. More specifically, the DWT can be viewed as a constant Q (center frequency/bandwidth) with octave spacing between the centers of the filters.

In the pyramidal algorithm, the signal is analyzed at different frequency bands with different resolutions for each band. This is achieved by successively decomposing the signal into a coarse approximation and detail information. The coarse approximation is then further decomposed using the same wavelet decomposition step. This decomposition step is achieved by successive

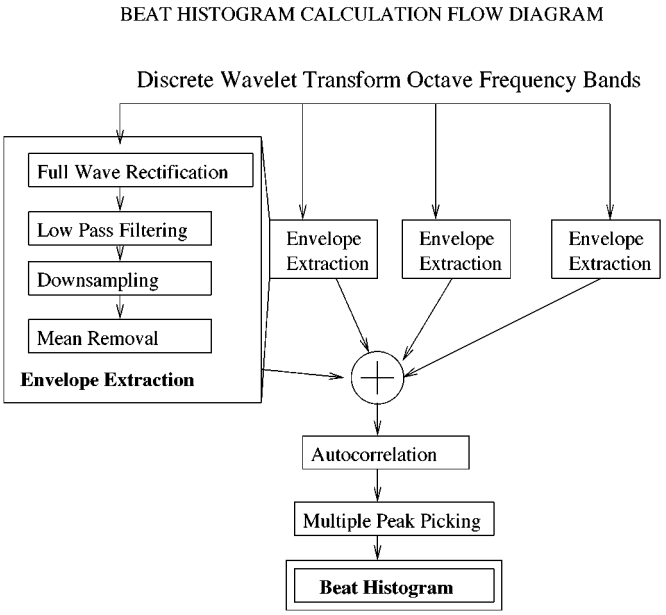

Fig. 1. Beat histogram calculation flow diagram.

highpass and lowpass filtering of the time domain signal and is defined by the following equations:

$$y_{high}[k] = \sum_n x[n]g[2k-n] \tag{5}$$

$$y_{low}[k] = \sum_n x[n]h[2k-n] \tag{6}$$

where $y_{high}[k]$, $y_{low}[k]$ are the outputs of the highpass (g) and lowpass (h) filters, respectively after subsampling by two. The DAUB4 filters proposed by Daubechies [21] are used.

The feature set for representing rhythm structure is based on detecting the most salient periodicities of the signal. Fig. 1 shows the flow diagram of the beat analysis algorithm. The signal is first decomposed into a number of octave frequency bands using the DWT. Following this decomposition, the time domain amplitude envelope of each band is extracted separately. This is achieved by applying full-wave rectification, low pass filtering, and downsampling to each octave frequency band. After mean removal, the envelopes of each band are then summed together and the autocorrelation of the resulting sum envelope is computed. The dominant peaks of the autocorrelation function correspond to the various periodicities of the signal's envelope. These peaks are accumulated over the whole sound file into a *beat histogram* where each bin corresponds to the peak lag, i.e., the beat period in beats-per-minute (bpm). Rather than adding one, the amplitude of each peak is added to the beat histogram. That way, when the signal is very similar to itself (strong beat) the histogram peaks will be higher.

The following building blocks are used for the beat analysis feature extraction.

1) Full Wave Rectification:

$$y[n] = |x[n]| \tag{7}$$

is applied in order to extract the temporal envelope of the signal rather than the time domain signal itself.

2) Low-Pass Filtering:

$$y[n] = (1-\alpha)x[n] + \alpha y[n-1] \tag{8}$$

i.e., a one-pole filter with an alpha value of 0.99 which is used to smooth the envelope. Full wave rectification followed by low-pass filtering is a standard envelope extraction technique.

3) Downsampling:

$$y[n] = x[kn] \tag{9}$$

where $k = 16$ in our implementation. Because of the large periodicities for beat analysis, downsampling the signal reduces computation time for the autocorrelation computation without affecting the performance of the algorithm.

4) Mean Removal:

$$y[n] = x[n] - E[x[n]] \tag{10}$$

is applied in order to make the signal centered to zero for the autocorrelation stage.

5) Enhanced Autocorrelation:

$$y[k] = \frac{1}{N}\sum_n x[n]x[n-k] \tag{11}$$

the peaks of the autocorrelation function correspond to the time lags where the signal is most similar to itself. The time lags of peaks in the right time range for rhythm analysis correspond to beat periodicities. The autocorrelation function is enhanced using a similar method to the multipitch analysis model of Tolonen and Karjalainen [22] in order to reduce the effect of integer multiples of the basic periodicities. The original autocorrelation function of the summary of the envelopes, is clipped to positive values and then time-scaled by a factor of two and subtracted from the original clipped function. The same process is repeated with other integer factors such that repetitive peaks at integer multiples are removed.

6) Peak Detection and Histogram Calculation: The first three peaks of the enhanced autocorrelation function that are in the appropriate range for beat detection are selected and added to a *beat histogram (BH)*. The bins of the histogram correspond to beats-per-minute (bpm) from 40 to 200 bpm. For each peak of the enhanced autocorrelation function the peak amplitude is added to the histogram. That way peaks that have high amplitude (where the signal is highly similar) are weighted more strongly than weaker peaks in the histogram calculation.

7) Beat Histogram Features: Fig. 2 shows a beat histogram for a 30-s excerpt of the song "Come Together" by the Beatles. The two main peaks of the BH correspond to the main beat at approximately 80 bpm and its first harmonic (twice the speed) at 160 bpm. Fig. 3 shows four beat histograms of pieces from different musical genres. The upper left corner, labeled classical, is the BH of an excerpt from "La Mer" by Claude Debussy. Because of the complexity of the multiple instruments of the orchestra there is no strong self-similarity and there is no clear dominant peak in the histogram. More strong peaks can be seen at the lower left corner, labeled jazz, which is an excerpt from a live performance by Dee Dee Bridgewater. The two peaks correspond to the beat of the song (70 and 140 bpm). The BH of Fig. 2 is shown on the upper right corner where the peaks are more pronounced because of the stronger beat of rock music.

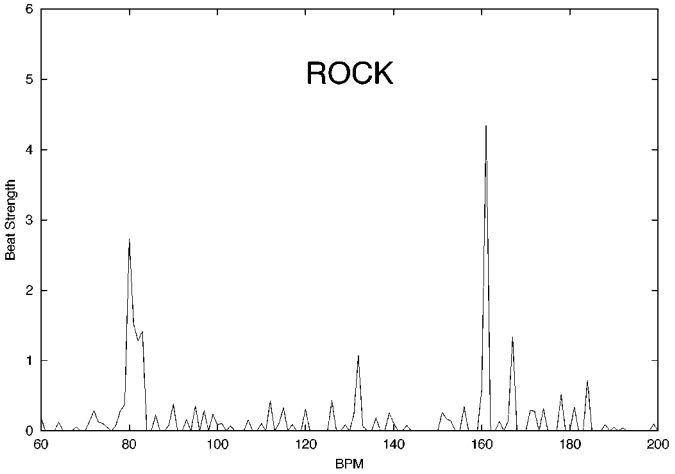

Fig. 2. Beat histogram example.

The highest peaks of the lower right corner indicate the strong rhythmic structure of a HipHop song by Neneh Cherry.

A small-scale study (20 excerpts from various genres) confirmed that most of the time (18/20) the main beat corresponds to the first or second BH peak. The results of this study and the initial description of beat histograms can be found in [23]. Unlike previous work in automatic beat detection which typically aims to provide only an estimate of the main beat (or tempo) of the song and possibly a measure of its strength, the BH representation captures more detailed information about the rhythmic content of the piece that can be used to intelligently guess the musical genre of a song. Fig. 3 indicates that the BH of different musical genres can be visually differentiated. Based on this observation a set of features based on the BH are calculated in order to represent rhythmic content and are shown to be useful for automatic musical genre classification. These are:

- **A0, A1**: relative amplitude (divided by the sum of amplitudes) of the first, and second histogram peak;
- **RA**: ratio of the amplitude of the second peak divided by the amplitude of the first peak;
- **P1, P2**: period of the first, second peak in bpm;
- **SUM**: overall sum of the histogram (indication of beat strength).

For the BH calculation, the DWT is applied in a window of 65 536 samples at 22 050 Hz sampling rate which corresponds to approximately 3 s. This window is advanced by a hop size of 32 768 samples. This larger window is necessary to capture the signal repetitions at the beat and subbeat levels.

D. Pitch Content Features

The pitch content feature set is based on multiple pitch detection techniques. More specifically, the multipitch detection algorithm described by Tolonen and Karjalainen [22] is utilized. In this algorithm, the signal is decomposed into two frequency bands (below and above 1000 Hz) and amplitude envelopes are extracted for each frequency band. The envelope extraction is performed by applying half-wave rectification and low-pass filtering. The envelopes are summed and an enhanced autocorrelation function is computed so that the effect of integer multiples of the peak frequencies to multiple pitch detection is reduced.

The prominent peaks of this summary enhanced autocorrelation function (SACF) correspond to the main pitches for that short segment of sound. This method is similar to the beat detection structure for the shorter periods corresponding to pitch perception. The three dominant peaks of the SACF are accumulated into a PH over the whole soundfile. For the computation of the PH, a pitch analysis window of 512 samples at 22 050 Hz sampling rate (approximately 23 ms) is used.

The frequencies corresponding to each histogram peak are converted to musical pitches such that each bin of the PH corresponds to a musical note with a specific pitch (for example A4 = 440 Hz). The musical notes are labeled using the MIDI note numbering scheme. The conversion from frequency to MIDI note number can be performed using

$$n = 12 \log_2 \frac{f}{440} + 69 \tag{12}$$

where f is the frequency in Hertz and n is the histogram bin (MIDI note number).

Two versions of the PH are created: a *folded* (FPH) and *unfolded* histogram (UPH). The unfolded version is created using the above equation without any further modifications. In the folded case, all notes are mapped to a single octave using

$$c = n \bmod 12 \tag{13}$$

where c is the folded histogram bin (pitch class or chroma value), and n is the unfolded histogram bin (or MIDI note number). The folded version contains information regarding the pitch classes or harmonic content of the music whereas the unfolded version contains information about the pitch range of the piece. The FPH is similar in concept to the chroma-based representations used in [24] for audio-thumbnailing. More information regarding the chroma and height dimension of musical pitch can be found in [25]. The relation of musical scales to frequency is discussed in more detail in [26].

Finally, the FPH is mapped to a circle of fifths histogram so that adjacent histogram bins are spaced a fifth apart rather than a semitone. This mapping is achieved by

$$c' = (7 \times c) \bmod 12 \tag{14}$$

where c' is the new folded histogram bin after the mapping and c is the original folded histogram bin. The number seven corresponds to seven semitones or the music interval of a fifth. That way, the distances between adjacent bins after the mapping are better suited for expressing tonal music relations (tonic-dominant) and the extracted features result in better classification accuracy.

Although musical genres by no means can be characterized fully by their pitch content, there are certain tendencies that can lead to useful feature vectors. For example jazz or classical music tend to have a higher degree of pitch change than rock or pop music. As a consequence, pop or rock music pitch histograms will have fewer and more pronounced peaks than the histograms of jazz or classical music.

Based on these observations the following features are computed from the UPH and FPH in order to represent pitch content.

- **FA0**: Amplitude of maximum peak of the folded histogram. This corresponds to the most dominant pitch class of the song. For tonal music this peak will typically

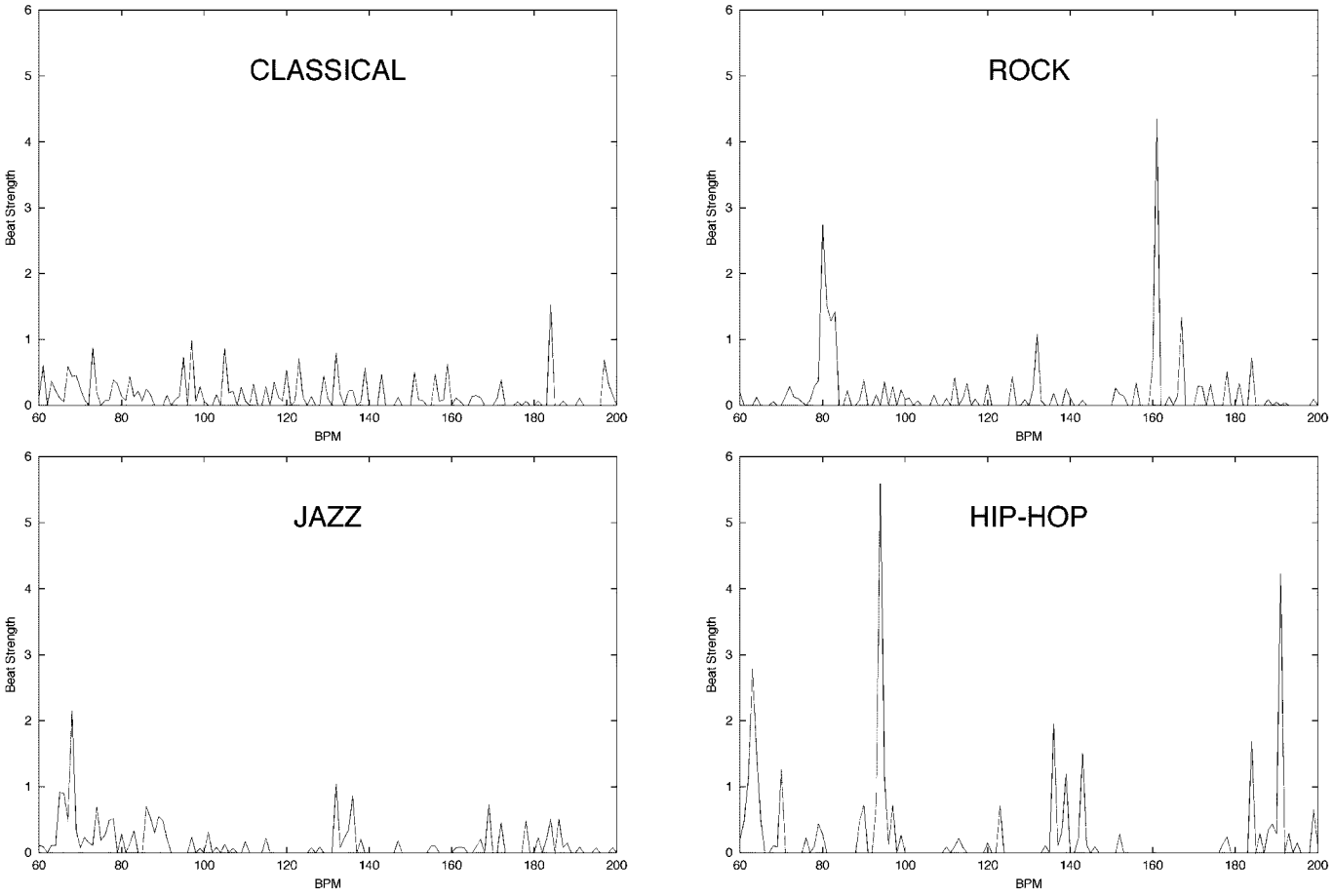

Fig. 3. Beat histogram examples.

correspond to the tonic or dominant chord. This peak will be higher for songs that do not have many harmonic changes.

- **UP0**: Period of the maximum peak of the unfolded histogram. This corresponds to the octave range of the dominant musical pitch of the song.
- **FP0**: Period of the maximum peak of the folded histogram. This corresponds to the main pitch class of the song.
- **IPO1**: Pitch interval between the two most prominent peaks of the folded histogram. This corresponds to the main tonal interval relation. For pieces with simple harmonic structure this feature will have value 1 or -1 corresponding to fifth or fourth interval (tonic-dominant).
- **SUM** The overall sum of the histogram. This is feature is a measure of the strength of the pitch detection.

E. Whole File and Real-Time Features

In this work, both the rhythmic and pitch content feature set are computed over the whole file. This approach poses no problem if the file is relatively homogeneous but is not appropriate if the file contains regions of different musical texture. Automatic segmentation algorithms [27], [28] can be used to segment the file into regions and apply classification to each region separately. If real-time performance is desired, only the timbral texture feature set can be used. It might possible to compute the rhythmic and pitch features in real-time using only short-time information but we have not explored this possibility.

IV. Evaluation

In order to evaluate the proposed feature sets, standard statistical pattern recognition classifiers were trained using real-world data collected from a variety of different sources.

A. Classification

For classification purposes, a number of standard statistical pattern recognition (SPR) classifiers were used. The basic idea behind SPR is to estimate the probability density function (pdf) for the feature vectors of each class. In supervised learning a labeled training set is used to estimate the pdf for each class. In the simple Gaussian (GS) classifier, each pdf is assumed to be a multidimensional Gaussian distribution whose parameters are estimated using the training set. In the Gaussian mixture model (GMM) classifier, each class pdf is assumed to consist of a mixture of a specific number K of multidimensional Gaussian distributions. The iterative EM algorithm can be used to estimate the parameters of each Gaussian component and the mixture weights. In this work GMM classifiers with diagonal covariance matrices are used and their initialization is performed using the K-means algorithm with multiple random starting points. Finally, the K-nearest neighbor (K-NN) classifier is an example

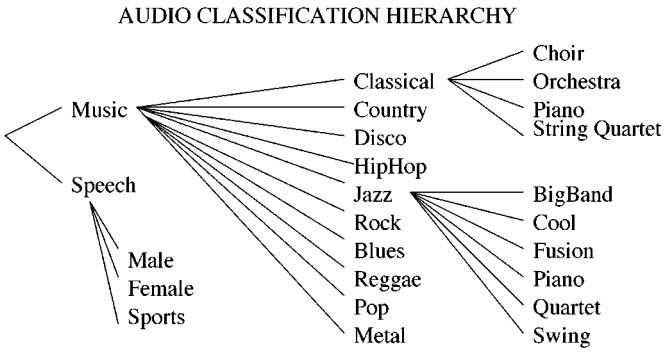

Fig. 4. Audio classification hierarchy.

TABLE I
CLASSIFICATION ACCURACY MEAN AND STANDARD DEVIATION

	Genres(10)	Classical(4)	Jazz(6)
Random	10	25	16
RT GS	44 ± 2	61 ± 3	53 ± 4
GS	59 ± 4	77 ± 6	61 ± 8
GMM(2)	60 ± 4	81 ± 5	66 ± 7
GMM(3)	61 ± 4	88 ± 4	68 ± 7
GMM(4)	61 ± 4	88 ± 5	62 ± 6
GMM(5)	61 ± 4	88 ± 5	59 ± 6
KNN(1)	59 ± 4	77 ± 7	57 ± 6
KNN(3)	60 ± 4	78 ± 6	58 ± 7
KNN(5)	56 ± 3	70 ± 6	56 ± 6

of a nonparametric classifier where each sample is labeled according to the majority of its K nearest neighbors. That way, no functional form for the pdf is assumed and it is approximated locally using the training set. More information about statistical pattern recognition can be found in [29].

B. Datasets

Fig. 4 shows the hierachy of musical genres used for evaluation augmented by a few (three) speech-related categories. In addition, a music/speech classifier similar to [4] has been implemented. For each of the 20 musical genres and three speech genres, 100 representative excerpts were used for training. Each excerpt was 30 s long resulting in (23 * 100 * 30 s = 19 h) of training audio data. To ensure variety of different recording qualities the excerpts were taken from radio, compact disks, and MP3 compressed audio files. The files were stored as 22 050 Hz, 16-bit, mono audio files. An effort was made to ensure that the training sets are representative of the corresponding musical genres. The Genres dataset has the following classes: classical, country, disco, hiphop, jazz, rock, blues, reggae, pop, metal. The classical dataset has the following classes: choir, orchestra, piano, string quartet. The jazz dataset has the following classes: bigband, cool, fusion, piano, quartet, swing.

C. Results

Table I shows the classification accuracy percentage results of different classifiers and musical genre datasets. With the exception of the RT GS row, these results have been computed using a single-vector to represent the whole audio file. The vector consists of the timbral texture features [9 (FFT) + 10 (MFCC) = 19 dimensions], the rhythmic content features (6 dimensions), and the pitch content features (five dimensions) resulting in a 30-dimensional feature vector. In order to compute a single timbral-texture vector for the whole file the mean feature vector over the whole file is used.

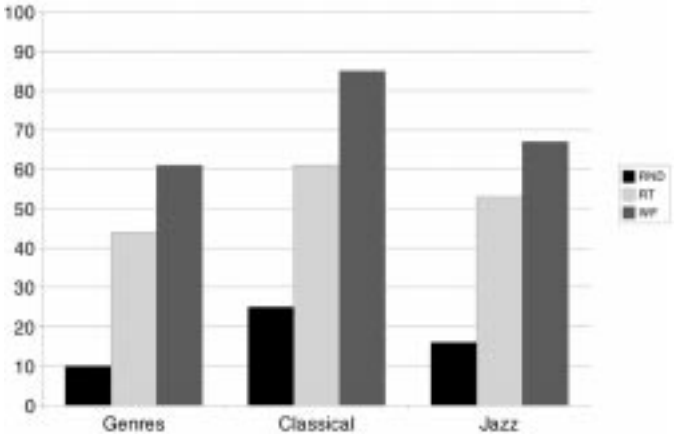

Fig. 5. Classification accuracy percentages (RND = random, RT = real time, WF = whole file).

The row RT GS shows classification accuracy percentage results for real-time classification per frame using only the timbral texture feature set (19 dimensions). In this case, each file is represented by a time series of feature vectors, one for each *analysis window*. Frames from the same audio file are never split between training and testing data in order to avoid false higher accuracy due to the similarity of feature vectors from the same file. A comparison of random classification, real-time features, and whole-file features is shown in Fig. 5. The data for creating this bar graph corresponds to the random, RT GS, and GMM(3) rows of Table I.

The classification results are calculated using a ten-fold cross-validation evaluation where the dataset to be evaluated is randomly partitioned so that 10% is used for testing and 90% is used for training. The process is iterated with different random partitions and the results are averaged (for Table I, 100 iterations were performed). This ensures that the calculated accuracy will not be biased because of a particular partitioning of training and testing. If the datasets are representative of the corresponding musical genres then these results are also indicative of the classification performance with real-world unknown signals. The $\pm$ part shows the standard deviation of classification accuracy for the iterations. The row labeled *random* corresponds to the classification accuracy of a chance guess.

The additional music/speech classification has 86% (random would be 50%) accuracy and the speech classification (male, female, sports announcing) has 74% (random 33%). Sports announcing refers to any type of speech over a very noisy background. The STFT-based feature set is used for the music/speech classification and the MFCC-based feature set is used for the speech classification.

1) Confusion Matrices: Table II shows more detailed information about the musical genre classifier performance in the form of a confusion matrix. In a confusion matrix, the columns correspond to the actual genre and the rows to the predicted genre. For example, the cell of row 5, column 1 with value 26 means that 26% of the classical music (column 1) was wrongly

TABLE II
GENRE CONFUSION MATRIX

	cl	co	di	hi	ja	ro	bl	re	po	me
cl	**69**	0	0	0	1	0	0	0	0	0
co	0	**53**	2	0	5	8	6	4	2	0
di	0	8	**52**	11	0	13	14	5	9	6
hi	0	3	18	**64**	1	6	3	26	7	6
ja	26	4	0	0	**75**	8	7	1	2	1
ro	5	13	4	1	9	**40**	14	1	7	33
bl	0	7	0	1	3	4	**43**	1	0	0
re	0	9	10	18	2	12	11	**59**	7	1
po	0	2	14	5	3	5	0	3	**66**	0
me	0	1	0	1	0	4	2	0	0	**53**

TABLE III
JAZZ CONFUSION MATRIX

	BBand	Cool	Fus.	Piano	4tet	Swing
BBand	**42**	2	1	0	6	1
Cool	21	**67**	5	4	23	10
Fus.	28	16	**88**	0	38	22
Piano	1	0	0	**80**	0	0
4tet	4	5	2	0	**19**	5
Swing	4	10	4	16	14	**62**

TABLE IV
CLASSICAL CONFUSION MATRIX

	Choir	Orch.	Piano	Str.4tet
Choir	**99**	7	7	3
Orch.	0	**58**	2	7
Piano	0	9	**86**	4
Str.4tet	1	26	5	**86**

classified as jazz music (row 2). The percentages of correct classification lie in the diagonal of the confusion matrix. The confusion matrix shows that the misclassifications of the system are similar to what a human would do. For example, *classical* music is misclassified as *jazz* music for pieces with strong rhythm from composers like Leonard Bernstein and George Gershwin. *Rock* music has the worst classification accuracy and is easily confused with other genres which is expected because of its broad nature.

Tables III and IV show the confusion matrices for the classical and jazz genre datasets. In the classical genre dataset, *orchestral* music is mostly misclassified as *string quartet*. As can be seen from the confusion matrix (Table III), jazz genres are mostly misclassified as *fusion*. This is due to the fact that *fusion* is a broad category that exhibits large variability of feature values. *jazz quartet* seems to be a particularly difficult genre to correctly classify using the proposed features (it is mostly misclassified as *cool* and *fusion*).

2) Importance of Texture Window Size: Fig. 6 shows how changing the size of the *texture window* affects the classification performance. It can be seen that the use of a *texture window* increases significantly the classification accuracy. The value of zero *analysis windows* corresponds to using directly the features computed from the *analysis window*. After approximately 40 *analysis windows* (1 s) subsequent increases in texture window size do not improve classification as they do not provide any additional statistical information. Based on this plot, the value of

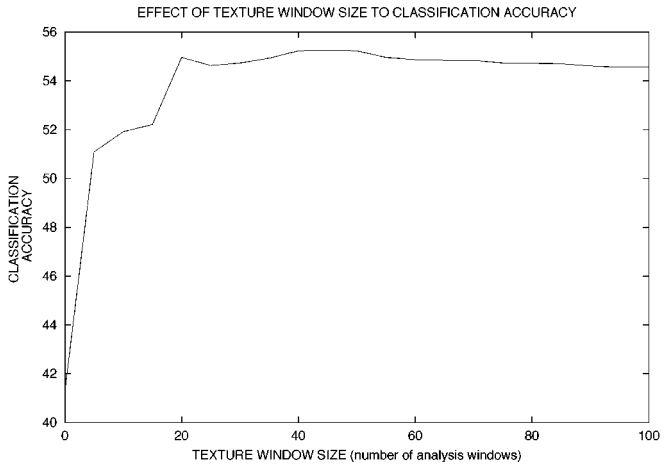

Fig. 6. Effect of texture window size to classification accuracy.

TABLE V
INDIVIDUAL FEATURE SET IMPORTANCE

	Genres	Classical	Jazz
RND	10	25	16
PHF(5)	23	40	26
BHF(6)	28	39	31
STFT(9)	45	78	58
MFCC(10)	47	61	56
FULL(30)	59	77	61

40 analysis windows was chosen as the *texture window* size. The timbral-texture feature set (STFT and MFCC) for the whole file and a single Gaussian classifier (GS) were used for the creation of Fig. 6.

3) Importance of Individual Feature Sets: Table V shows the individual importance of the proposed feature sets for the task of automatic musical genre classification. As can be seen, the nontimbral texture features pitch histogram features (PHF) and beat histogram features (BHF) perform worse than the timbral-texture features (STFT, MFCC) in all cases. However, in all cases, the proposed feature sets perform better than random classification therefore provide some information about musical genre and therefore musical content in general. The last row of Table V corresponds to the full combined feature set and the first row corresponds to random classification. The number in parentheses beside each feature set denotes the number of individual features for that particular feature set. The results of Table V were calculated using a single Gaussian classifier (GS) using the whole-file approach.

The classification accuracy of the combined feature set, in some cases, is not significantly increased compared to the individual feature set classification accuracies. This fact does not necessarily imply that the features are correlated or do not contain useful information because it can be the case that a specific file is correctly classified by two different feature sets that contain different and uncorrelated feature information. In addition, although certain individual features are correlated, the addition of each specific feature improves classification accuracy. The rhythmic and pitch content feature sets seem to play a less important role in the classical and jazz dataset classification compared to the Genre dataset. This is an indication that it is possible

TABLE VI
BEST INDIVIDUAL FEATURES

	Genres
BHF.SUM	20
PHF.FP0	23
STFT.VCTRD	29
MFCC.MMFCC1	25

that genre-specific feature sets need to be designed for more detailed subgenre classification.

Table VI shows the best individual features for each feature set. These are the sum of the beat histogram (BHF.SUM), the period of the first peak of the folded pitch histogram (PHF.FP0), the variance of the spectral centroid over the texture window (STFT.FPO) and the mean of the first MFCC coefficient over the texture window (MFCC.MMFCC1).

D. Human Performance for Genre Classification

The performance of humans in classifying musical genre has been investigated in [30]. Using a ten-way forced-choice paradigm college students were able to accurately judge (53% correct) after listening to only 250-ms samples and (70% correct) after listening to 3 s (chance would be 10%). Listening to more than 3 s did not improve their performance. The subjects where trained using representative samples from each genre. The ten genres used in this study were: blues, country, classical, dance, jazz, latin, pop, R&B, rap, and rock. Although direct comparison of these results with the automatic musical genre classification results, is not possible due to different genres and datasets, it is clear that the automatic performance is not far away from the human performance. Moreover, these results indicate the fuzzy nature of musical genre boundaries.

V. CONCLUSIONS AND FUTURE WORK

Despite the fuzzy nature of genre boundaries, musical genre classification can be performed automatically with results significantly better than chance, and performance comparable to human genre classification. Three feature sets for representing timbral texture, rhythmic content and pitch content of music signals were proposed and evaluated using statistical pattern recognition classifiers trained with large real-world audio collections. Using the proposed feature sets classification of 61% (nonreal time) and 44% (real time), has been achieved in a dataset consisting of ten musical genres. The success of the proposed features for musical genre classification testifies to their potential as the basis for other types of automatic techniques for music signals such as similarity retrieval, segmentation and audio thumbnailing which are based on extracting features to describe musical content.

An obvious direction for future research is expanding the genre hierarchy both in width and depth. Other semantic descriptions such as emotion or voice style will be investigated as possible classification categories. More exploration of the pitch content feature set could possibly lead to better performance. Alternative multiple pitch detection algorithms, for example based on cochlear models, could be used to create the pitch histograms. For the calculation of the beat histogram we plan to explore other filterbank front-ends as well as onset based periodicity detection as in [14] and [15]. We are also planning to investigate real-time running versions of the rhythmic structure and harmonic content feature sets. Another interesting possibility is the extraction of similar features directly from MPEG audio compressed data as in [31] and [32]. We are also planning to use the proposed feature sets with alternative classification and clustering methods such as artificial neural networks. Finally, we are planning to use the proposed feature set for query-by-example similarity retrieval of music signals and audio thumbnailing. By having separate feature sets to represent timbre, rhythm, and harmony, different types of similarity retrieval are possible. Two other possible sources of information about musical genre content are melody and singer voice. Although melody extraction is a hard problem that is not solved for general audio it might be possible to obtain some statistical information even from imperfect melody extraction algorithms. Singing voice extraction and analysis is another interesting direction for future research.

The software used for this paper is available as part of MARSYAS [33], a free software framework for rapid development and evaluation of computer audition applications. The framework follows a client–server architecture. The C++ server contains all the pattern recognition, signal processing, and numerical computations and is controlled by a client graphical user interface written in Java. MARSYAS is available under the GNU Public License at http://www.cs.princeton.edu/~gtzan/marsyas.html.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their careful reading of the paper and suggestions for improvement. D. Turnbull helped with the implementation of the GenreGram user interface and G. Tourtellot implemented the multiple pitch analysis algorithm. Many thanks to G. Essl for discussions and help with the beat histogram calculation.

REFERENCES

[1] F. Pachet and D. Cazaly, "A classification of musical genre," in *Proc. RIAO Content-Based Multimedia Information Access Conf.*, Paris, France, Mar. 2000.
[2] S. Davis and P. Mermelstein, "Experiments in syllable-based recognition of continuous speech," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 28, pp. 357–366, Aug. 1980.
[3] J. Saunders, "Real time discrimination of broadcast speech/music," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, 1996, pp. 993–996.
[4] E. Scheirer and M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, 1997, pp. 1331–1334.
[5] D. Kimber and L. Wilcox, "Acoustic segmentation for audio browsers," in *Proc. Interface Conf.*, Sydney, Australia, July 1996.
[6] T. Zhang and J. Kuo, "Audio content analysis for online audiovisual data segmentation and classification," *Trans. Speech Audio Processing*, vol. 9, pp. 441–457, May 2001.
[7] A. L. Berenzweig and D. P. Ellis, "Locating singing voice segments within musical signals," in *Proc. Int. Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)* Mohonk, NY, 2001, pp. 119–123.
[8] E. Wold, T. Blum, D. Keislar, and J. Wheaton, "Content-based classification, search, and retrieval of audio," *IEEE Multimedia*, vol. 3, no. 2, 1996.

[9] J. Foote, "Content-based retrieval of music and audio," *Multimed. Storage Archiv. Syst. II*, pp. 138–147, 1997.
[10] G. Li and A. Khokar, "Content-based indexing and retrieval of audio data using wavelets," in *Proc. Int. Conf. Multimedia Expo II*, 2000, pp. 885–888.
[11] S. Li, "Content-based classification and retrieval of audio using the nearest feature line method," *IEEE Trans. Speech Audio Processing*, vol. 8, pp. 619–625, Sept. 2000.
[12] E. Scheirer, "Tempo and beat analysis of acoustic musical signals," *J. Acoust. Soc. Amer.*, vol. 103, no. 1, p. 588, 601, Jan. 1998.
[13] M. Goto and Y. Muraoka, "Music understanding at the beat level: Real-time beat tracking of audio signals," in *Computational Auditory Scene Analysis*, D. Rosenthal and H. Okuno, Eds. Mahwah, NJ: Lawrence Erlbaum, 1998, pp. 157–176.
[14] J. Laroche, "Estimating tempo, swing and beat locations in audio recordings," in *Proc. Int. Workshop on Applications of Signal Processing to Audio and Acoustics WASPAA*, Mohonk, NY, 2001, pp. 135–139.
[15] J. Seppänen, "Quantum grid analysis of musical signals," in *Proc. Int. Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)* Mohonk, NY, 2001, pp. 131–135.
[16] J. Foote and S. Uchihashi, "The beat spectrum: A new approach to rhythmic analysis," in *Proc. Int. Conf. Multimedia Expo.*, 2001.
[17] G. Tzanetakis, G. Essl, and P. Cook, "Automatic musical genre classification of audio signals," in *Proc. Int. Symp. Music Information Retrieval (ISMIR)*, Oct. 2001.
[18] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
[19] B. Logan, "Mel frequency cepstral coefficients for music modeling," in *Proc. Int. Symp. Music Information Retrieval (ISMIR)*, 2000.
[20] S. G. Mallat, *A Wavelet Tour of Signal Processing*. New York: Academic, 1999.
[21] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Commun. Pure Appl. Math*, vol. 41, pp. 909–996, 1988.
[22] T. Tolonen and M. Karjalainen, "A computationally efficient multipitch analysis model," *IEEE Trans. Speech Audio Processing*, vol. 8, pp. 708–716, Nov. 2000.
[23] G. Tzanetakis, G. Essl, and P. Cook, "Audio analysis using the discrete wavelet transform," in *Proc. Conf. Acoustics and Music Theory Applications*, Sept. 2001.
[24] M. A. Bartsch and G. H. Wakefield, "To catch a chorus: Using chroma-based representation for audio thumbnailing," in *Proc. Int. Workshop on Applications of Signal Processing to Audio and Acoustics* Mohonk, NY, 2001, pp. 15–19.
[25] R. N. Shepard, "Circularity in judgments of relative pitch," *J. Acoust. Soc. Amer.*, vol. 35, pp. 2346–2353, 1964.
[26] J. Pierce, "Consonance and scales," in *Music Cognition and Computerized Sound*, P. Cook, Ed. Cambridge, MA: MIT Press, 1999, pp. 167–185.
[27] J.-J. Aucouturier and M. Sandler, "Segmentation of musical signals using hidden Markov models," in *Proc. 110th Audio Engineering Society Convention*, Amsterdam, The Netherlands, May 2001.
[28] G. Tzanetakis and P. Cook, "Multifeature audio segmentation for browsing and annotation," in *Proc. Workshop Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, 1999.
[29] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. New York: Wiley, 2000.
[30] D. Perrot and R. Gjerdigen, "Scanning the dial: An exploration of factors in identification of musical style," in *Proc. Soc. Music Perception Cognition*, 1999, p. 88, (abstract).
[31] D. Pye, "Content-based methods for the management of digital music," in *Proc. Int. Conf Acoustics, Speech, Signal Processing (ICASSP)*, 2000.
[32] G. Tzanetakis and P. Cook, "Sound analysis using MPEG compressed audio," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Istanbul, Turkey, 2000.
[33] ——, "Marsyas: A framework for audio analysis," *Organized Sound*, vol. 4, no. 3, 2000.

George Tzanetakis (S'98) received the B.Sc. degree in computer science from the University of Crete, Greece, and the M.A. degree in computer science from Princeton University, Princeton, NJ, where he is currently pursuing the Ph.D. degree.

His research interests are in the areas of signal processing, machine learning, and graphical user interfaces for audio content analysis with emphasis on music information retrieval.

Perry Cook (S'84–M'90) received the B.A. degree in music from the University of Missouri at Kansas City (UMKC) Conservatory of Music, the B.S.E.E. degree from UMKC Engineering School, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA.

He is Associate Professor of computer science, with a joint appointment in music, at Princeton University, Princeton, NJ. He served as Technical Director for Stanford's Center for Computer Research in Music and Acoustics and has consulted and worked in the areas of DSP, image compression, music synthesis, and speech processing for NeXT, Media Vision, and other companies. His research interests include physically based sound synthesis, human–computer interfaces for the control of sound, audio analysis, auditory display, and immersive sound environments.