EVALUATION OF FREQUENTLY USED AUDIO FEATURES FOR CLASSIFICATION OF MUSIC INTO PERCEPTUAL CATEGORIES

Tim Pohle[1], *Elias Pampalk*[1] *and Gerhard Widmer*[1,2]

[1]Austrian Research Institute for Artificial Intelligence, Vienna
[2]Department of Computational Perception, Johannes Kepler University Linz
{tim, elias, gerhard}@oefai.at

ABSTRACT

The ever-growing amount of available music induces an increasing demand for Music Information Retrieval (*MIR*) applications such as music recommendation applications or automatic classification algorithms.

When audio-based, a crucial part of such systems are the audio feature extraction routines. In this paper, we evaluate how well a variety of combinations of feature extraction and machine learning algorithms are suited to classify music into perceptual categories. The examined categorizations are perceived tempo, mood (happy / neutral /sad), emotion (soft / neutral / aggressive), complexity, and vocal content.

The aim is to contribute to the investigation which aspects of music are not captured by the common audio descriptors; from our experiments we can conclude that most of the examined categorizations are not captured well. This indicates that more research is needed on alternative (possibly extra-musical) sources of information for useful music classification.

1. INTRODUCTION

Music Information Retrieval (*MIR*) deals with the automatic extraction of useful information from music that is given as audio data. The outcome of MIR research could be useful for a variety of applications; most notably, the ongoing change in music distribution that gradually shifts sales to online music stores makes reliable automatic music recommendation and classification systems desirable. Also, MIR algorithms could help in radio or DJ playlist generation, and the organization of personal and public music collections.

Much research has already been done in the field of audio-based music similarity measures, and automatic classification of music that is given as audio. Most often, these algorithms are evaluated relative to the task of genre classification (in some publications, also other categories are considered, e.g. if the pieces are on the same album or if they are from the same artist). Interestingly, in most publications it is not evaluated which different aspects of music are captured by the applied feature extraction algorithms. In this paper, we evaluate how well several combinations of well-known feature extraction and machine learning algorithms are suited for capturing aspects of mood, emotion, vocal content, perceived tempo, and complexity. An enhanced knowledge about this topic could support the development of improved feature extraction algorithms that are also able to describe the facets of music that are not captured by the existing ones.

2. RELATED WORK

Substantial research has recently been carried out on the problem of *automatic genre classification* (e.g., [21, 3, 10], to name but a few). There has even been a systematic Genre Classification Contest at the last International Conference on Music Information Retrieval (ISMIR'04) [1]. While by far the most work has been spent on classifying music according to genre, there have also been some isolated attempts at applying other labellings and categorizations that are intrinsic to music.

An algorithm to automatically detect mood from classical music given as acoustic data is presented in [15].

[14] apply a 30-attribute set and a Support Vector Machine for classification of music into 13 emotion classes; the system has an "overall low performance": for example, for micro-averaging (i.e. per-category weighted averaging) the recall is 0.36, and the precision is 0.59.

A system that automatically learns relations between adjectives and audio features is presented in [23]. The adjectives are gained by crawling the web. As the web-based data is given per artist, the audio features of several tracks of each artist are also combined, and the associations between adjectives and audio features are made at the artist level.

In [6], a system is presented that learns labels related to mood and genre, and gives probabilities for the occurrence of each label for unseen instances. This system was eval-

[1]http://ismir2004.ismir.net/genre_contest/index.htm

uated using one hundred different descriptive labels, with the pieces belonging to one of the two major classes *Rock* or *Electronica*. The labellings that were obtained from the web were also given at the artist level.

Perceived Complexity has multiple dimensions; there is research ongoing to develop a feature extraction algorithm that takes them into account ([20]).

In [17], an interesting approach is presented for automatic development of new features that describe almost arbitrary aspects of music.

In this publication, we systematically evaluate features that are used frequently in the literature, with labels given at the song level.

3. EXPERIMENTAL SETUP

To estimate which of the examined categorizations of music are amenable to automatic classification, and which features that can be extracted from audio are useful in this task, the following approach is used:

First, we calculate a wide range of features that have commonly been used in the literature on genre classification from the songs of a music collection, which were labeled according to the desired categories. These features were converted into attributes that are suited to be fed into standard machine learning algorithms. Three different attribute sets are evaluated in combination with twelve different machine learning algorithms. From these experiments, confusion matrices and overall classification accuracies are assessed. Each of the steps is described in detail in the next sections.

3.1. Song Collection and Target Categorizations

For the experiments, an in-house database consisting of 834 pieces given in mp3 format was used ([19]). The pieces were hand-labeled into the categories given in table 1 by one male subject. In this table, also the cardinalities of each class are given; the fact that they are unequally distributed is a consequence of taking a real-world song collection.

Except for genre[2] and focus, the categories are based on a common sense understanding: For example, the complexity can be affected by the kind of rhythm, by the melody line, or by the number of different instruments that appear. We are aware that for these categorizations (including genre), it is by no means granted that every human will categorize each of the songs in the same way; it is even not clear if one individual would choose the same categories when the labeling is repeated. However, asking humans is the only way to get the labellings.

[2]taken from *All Music Guide* (www.allmusic.com) and the genre descriptors of the *ID3 tagging system* (www.id3.org)

The categorizations have been chosen to reflect important dimensions of music that are not necessarily correlated (for example, it does not depend on the genre if a piece contains vocals). For a MIR system (e.g. a music recommendation system) it would be valuable to be able to distinguish between these categories, as they are an important part of how humans refer to music (e.g. "this band plays mostly fast and complex instrumental music").

Categorization	Classes (# of songs in class)
mood	happy (29%), neutral (50%), sad (21%)
perceived tempo	very slow (4%), slow (20%), medium (43%), fast (24%), very fast (5%), varying (4%)
complexity	low (18%), medium (56%), high (7%)
emotion	soft (29%), neutral (44%), aggressive (26%)
focus	vocals (6%), both (69%), instruments (26%)
genre	blues (1%), classical (5%), electronica (13%), folk (2%), jazz (1%), new age (5%), noise (0.1%), rock (60%), world (10%)

Tab. 1. Categorizations of the song collection used.

3.2. Feature Extraction

To extract features from the music, each piece was converted to wav format, downsampled to 11025 kHz mono, and 30 seconds exactly from the middle of it were taken to compute the features. For some of the features, the audio excerpt was divided into frames of 256 samples length, that overlapped by one half. The three tested feature sets are described in the next sections.

3.2.1. Set from [21]

This set was a vector containing 30 components for a song; they were implemented following the information and definitions given in [21].

- *Timbral Texture Features.* For each audio frame, the following values are extracted:
  - Spectral Centroid: the center of the magnitude distribution of the spectrum.
  - Spectral Rolloff: the frequency under which 95% of the power distribution is concentrated.

  - Spectral Flux, a measure of short-time changes of the spectrum.
  - Zero Crossing Rate, the number of times the time-domain signal passes the zero-level.
  - The first five MFCCs. MFCCs give an description of the envelope of the frame's spectrum.

  The timbral texture features are the mean and variance of these values of all frames, and Low Energy Rate. Altogether, this results in 19 values: one value for Low Energy Rate, two values (i.e. mean and variance) for each of Spectral Centroid, Spectral Rolloff, Spectral Flux, Zero Crossing rates, and ten values for the MFCCs.

- A beat histogram describes how much periodicity is in the audio excerpt at different tempo levels; in many cases, the most prominent peak corresponds to the main tempo of the excerpt. The *Rhythmic Content Features* are the following six properties of the beat histogram of the audio excerpt (following [21], where they are not motivated in detail):
  - *A0, A1:* relative amplitude of the two highest peaks (i.e. they are divided by the sum of the histogram),
  - *RA:* relative ratio of height of the second highest peak to the highest peak
  - *P1, P2:* bpm values of the two highest peaks
  - *SUM:* sum of all histogram bins, which is an indication of beat strength.
- In an analogous manner, into a pitch histogram describes how much each pitch height is present in the audio excerpt: the unfolded pitch histogram gives these values over the whole pitch range, while in the folded pitch histogram, all values that lie whole octaves apart are summed, resulting in a histogram that has 12 bins. The *Pitch Content Features* are five properties of the folded and unfolded pitch histograms:
  - Amplitude of the highest peak of the folded histogram, which "will be higher for songs that do not have many harmonic changes" ([21]).
  - Pitches of the highest peaks of the folded and unfolded histograms. For the unfolded histogram, this indicates the octave range of the dominant pitch of the piece; for the folded histogram, it indicates the main pitch class.
  - Interval between the two highest peaks of the folded histogram, which is related to the main interval region of the piece.
  - Sum of all histogram bins. This value is higher if the pitch was detected accurately.

Our results based on these features will not be directly comparable to the results in [21], because in [21], the procedure for selecting "representative" excerpts from pieces is not precisely specified.

3.2.2. Set made from some Mpeg7-LLDs

The Mpeg7 standard was designed to offer a comprehensive framework for describing the content of multimedia files. It offers techniques for metadata handling and for extracting features from different types of multimedia files. A part of the latter are features to describe basic properties of audio, called Low Level Descriptors (LLDs).

From the Mpeg7 LLDs given in [1], we selected a subset, as the others did not seem suitable for our specific type of experiments. For example, the Audio Waveform Descriptor was not used, as its main purpose is to support to display the audio envelope. Also, the *timbre descriptors* were not used, as they are best fitted to work on monophonic audio segments. The Mpeg7 LLDs that we used in our setup were:

- Audio Power, a measure for the power that is contained in the time-domain signal.
- Audio Spectrum Centroid, an analogue to the Spectral Centroid, but defined on a logarithmic frequency scale.
- Audio Spectrum Spread
- Audio Spectrum Flatness (on four bands), which measures how far the spectrum deviates from a flat curve in the respective frequency band.
- Audio Harmonicity, consisting of three values per frame:
  - Fundamental Frequency, an approximation of the main pitch.
  - Harmonicity Ratio, a measure how harmonic the current frame sounds.
  - Upper Limit of Harmonicity, i.e. the frequency above which the sound is not harmonic anymore.

Analogous to the set from [21], the attribute values for a song are the means and variances of the values of each time series or band, respectively, which gives a total of 20 attributes for each song.

3.2.3. "Large" set

The third set consisted of the 50 attributes from the other two sets, and additional attributes that were derived from the following features, because these features have been used

and reported by various authors in the context of music classification:

- a measure for the signal bandwidth of the current frame (e.g. [13]),
- Spectral Power ([26, 16]), which is an alternative measure for the signal energy,
- First three Statistical Moments of the frequency power distribution ([7]), with the first one being the Spectral Centroid,
- beat tracking and beat onset features, summarized in Inter Onset Interval Histograms (IOIHs, e.g. [9, 8]),
- a feature that estimates how "percussive" a piece is (one value per piece);
- a monophonic melody estimation, from which the average pitch height, the standard deviation of pitch height, the average note duration, and the relation of upward to downward pitch changes were calculated as attributes.

From the two types of IOIHs, from the Beat Histogram, and from the (folded and unfolded) Pitch Histograms the mean, the maximum and the standard deviation of the values appearing in each of five quantiles were chosen as attributes. From Bandwidth, Spectral Power and the Statistical Moments, mean and variance were taken as attributes.

Altogether, this resulted in 146 attributes that were calculated from this "large" feature set.

3.2.4. Proposed Algorithm from [3]

Additionally, the algorithm described in [2, 3] was used with the parameter set proposed in [3]. As this algorithm is a similarity function (i.e. its result is a distance measure rather than a set of attributes), only K-Nearest Neighbors classification was used for each of the categorizations. Calculation of this algorithm was done with the MA-Toolbox ([18]).

3.3. Machine Learning Algorithms

To evaluate the amount of information that is captured by the features, we use the feature data (*features*) to train (respectively apply) machine learning algorithms. Machine learning algorithms are built to "discover" the underlying structure in the feature data, and to extract information from it that is useful for the classification task at hand; so the rate of their success is an indicator for the quality of the features. The rate of success can be drawn from the average classification accuracy, which should be clearly higher than the baseline (which is the classification accuracy that is obtained when always the most frequent class is assigned).

As different machine learning algorithms use different approaches, they do not work equally well on all kinds of data; that is the reason why we used twelve variations of machine learning algorithms (all from the machine learning toolbox WEKA [25]):

- K-Nearest Neighbors, with $K \in \{1, 3, 5, 10\}$,
- Naïve Bayes, additional with a kernel estimation algorithm,
- the C4.5 algorithm (a decision tree learner),
- a Support Vector Machine (SVM),
- AdaBoost with C4.5 and with Decision Stump,
- Classification via Regression, applying M5 and linear regression.

Evaluation was done using a 10-fold cross validation (i.e., the machine learning algorithm was trained using the known class membership of 90% of the pieces in the collection, and the trained algorithm was used to classify the remaining 10%; this procedure was repeated ten times, so that each piece has been classified once).

4. RESULTS

Altogether, 240 combinations of attribute set / learning algorithm and categorization were evaluated. To get an overall picture of the results, for each categorization only the average classification accuracy of the best performing learning algorithm is given in the respective table, and only the interesting confusion matrices are depicted.

4.1. Focus

For the *focus* categorization, in most experiments the overall accuracy was below the baseline; table 2 shows that also the best classification accuracies do not indicate that the tested descriptor sets are suited for detection of vocals.

Focus Classification Results

Baseline	68.92 %
Set from [21]	71.08 %
Some Mpeg7 LLDs	70.00 %
"Large" set	71.20 %
Best from [3]	75.18 %

Tab. 2. Best classification accuracies for the focus categories (vocals / both / instruments).

In only four experiments the confusion matrices indicated an ability to distinguish between *vocal* pieces and pieces

that had the focus *both* (two of them are depicted in figure 1). Interestingly, each descriptor set was represented in these cases, and in all cases the learning algorithm was Naive Bayes (which produced classification accuracies below the baseline in all cases).

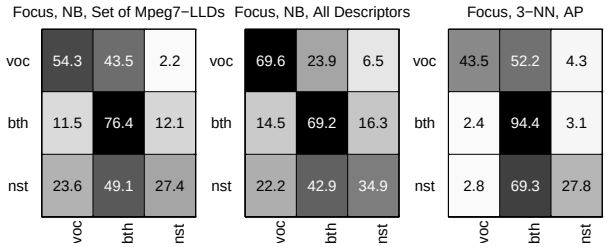

Fig. 1. Confusion matrices for the focus classes (*voc* - vocal, *bth* - both, *nst* - instrumental): each row gives the percentages of pieces belonging to the "true" category that are classified into the "predicted" categories. "True" categories are denoted at the left, "predicted" categories at the bottom. Left figure: Naive Bayes classification for the set made from Mpeg7-LLDs and for the set consisting of all implemented descriptors; for comparison, the right figure shows the confusion matrix of the best method from [3], which achieved the highest overall accuracy.

This outcome could be explained by the fact that no dedicated vocal feature was used; presumably, the use of a dedicated vocal detection algorithm (such as e.g. proposed in [5, 11]) would yield better results, especially for the discrimination between instrumental pieces and pieces that contain both instruments and vocals.

4.2. Complexity

For complexity, classification results are similar to those of the focus categorization: in most experiments, the baseline is not reached, and also the highest accuracies (summarized in table 3) are only somewhat over the baseline.

Complexity Classification Results

Baseline	75.66 %
Set from [21]	76.63 %
Some Mpeg7 LLDs	76.14 %
"Large" Set	76.87 %
Best from [3]	78.55 %

Tab. 3. Best classification accuracies for the complexity categories (low / medium / high).

The confusion matrices confirm the impression of poor performance: only pieces that belonged to the most frequent complexity class (*medium*) were correctly classified with high accuracy, which could easily be achieved by guessing the most frequent class. In all experiments, pieces with *low* and *high* complexity were more often misclassified than correctly classified, and no indications for class separability abilities are given.

The notion of 'complexity' as used here is probably too ill-defined to be acquired by a machine learning algorithm.

4.3. Perceived Tempo

For the tempo classification accuracies, the same picture appears (table 4): only in few cases the baseline is reached, and also the best accuracies are not much higher than the baseline.

Perceived Tempo Classification Results

Baseline	42.53 %
Set from [21]	42.53 %
Some Mpeg7 LLDs	43.13 %
"Large" Set	44.70 %
Best from [3]	48.67 %

Tab. 4. Best classification accuracies for the perceived tempo categories (very slow / slow / medium / fast / very fast / varying).

All but two confusion matrices indicate that the most frequent class can not be clearly discriminated from the other classes with the respective attribute set / learning algorithm combination.

The two confusion matrices that do not show a dark row at the position of the most frequent class are depicted in figure 2. With the set containing all implemented descriptors, Naive Bayes is able to classify pieces with *very slow*, *very fast*, and *varying* perceived tempo more accurately than the other learning algorithms. Pieces with *slow*, *medium* and *fast* are likely to be confused with *varying* perceived tempo.

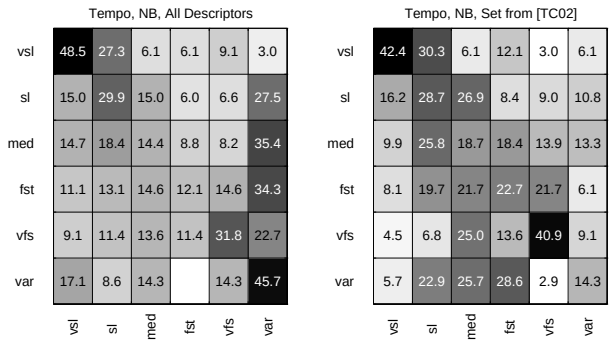

Fig. 2. Confusion matrices for Naive Bayes classification of the tempo classes (*vsl* - very slow, *sl* - slow, *med* - medium, *fst* - fast, *vfs* - very fast, *var* - varying) for the set consisting of all descriptors, and for the set from [21].

When only the descriptors contained in the set from [21] are used, the confusion matrix has confusions appearing

most frequently between neighboring classes (except for the *varying* class).

Maybe the perceived tempo depends on a rather complicated combination of several aspects: e.g. if the overall ground beat is slow, the perceived tempo could nonetheless be high, if only one instrument plays a quick melody; it also might depend on the "groove".

4.4. Emotion

From table 5, it can be seen that the best overall accuracies for the "emotion" categorization were achieved with the method from [3]. For the other descriptor set / learning algorithm combinations, overall accuracies were below the baseline in most cases, and also the highest accuracies that they achieved were clearly below the accuracies of the method from [3].

Emotion Classification Results

Baseline	44.46 %
Set from [21]	45.06 %
Some Mpeg7 LLDs	46.75 %
"Large" Set	47.47 %
Best from [3]	57.95 %

Tab. 5. Best classification accuracies for the emotion categories (soft / neutral / aggressive).

The main difference between the method from [3] and the other algorithms is that the method from [3] primarily aims at modeling timbral similarity, while the other methods incorporate concepts that are meant to describe also other aspects of music, such as melodic or harmonic content. Thus, the results suggest a high correlation of the "emotion" categories *soft*, *neutral*, and *aggressive* with timbre. In addition, a low correlation of other aspects of music (e.g. harmony, melody, rhythm) with these categories can be supposed, as different interpretations of the same piece can be soft or aggressive.

4.5. Mood

From table 6, it can be seen that the examined descriptor sets seem not to be suited for distinguishing between *happy*, *neutral* and *sad* songs.

Also, most confusion matrices show the black row indicating a lack of class separation ability; the two that deviate most from this appearance are again confusion matrices of Naive Bayes classifications (given in figure 3). They indicate that it might be possible to distinguish between *sad* and *happy* songs to some extent. As they are based on the set from [21] and on the set made from Mpeg7-LLDs, a feature selection might be interesting also in this case.

Mood Classification Results

Baseline	50.00 %
Set from [21]	50.00 %
Some Mpeg7 LLDs	50.00 %
"Large" Set	51.08 %
Best from [3]	50.24 %

Tab. 6. Best classification accuracies for the Mood categories (happy / neutral / sad).

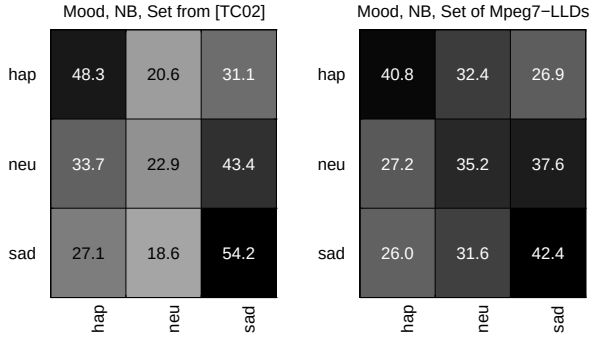

Fig. 3. Confusion matrices for Naive Bayes classification of the mood classes (*hap* - happy, *neu* - neutral, *sad* - sad) for the set made from Mpeg7-LLDs, and for the set from [21].

Presumably, the categorization into *happy*, *neutral* and *sad* pieces is too abstract and too subtle to be recognized by an algorithm.

4.6. Genre

The best genre classification accuracies for each descriptor set are given in table 7. These results are difficult to compare to those found in the literature, as the databases are different; but in consideration of the unequally distributed classes (and a high baseline of 60.48 %), they seem to be reasonable.

Genre Classification Results

Baseline	60.48 %
Set from [21]	65.66 %
Some Mpeg7 LLDs	64.94 %
"Large" Set	69.52 %
Best from [3]	70.84 %

Tab. 7. Best classification accuracies for the nine genre categories.

5. CONCLUSIONS

The most important observations are:

- Overall results were hardly over the baseline for most of the feature set / learning algorithm combinations for most of the examined categorizations.
- In all cases, the best accuracy values obtained from the proposed algorithm from [3] were better than the best accuracy values obtained from the other three feature sets.
- From the examined categorizations, the algorithms generally seem to work best for the *Genre* categorization, which is the categorization that is most frequently used in the literature. An exception is the proposed algorithm from [3] which also worked comparably for the *Emotion* categorization.

In more detail, for the individual categorizations we got the following results:

- No indication was found that the most commonly used audio features are useful for classification of pieces into the *Complexity* classes.
- For the *Focus*, *Perceived Tempo*, and *Mood* categorizations, overall classification accuracies were mostly below the baseline. Nevertheless, some confusion matrices indicate that the examined features might capture some aspects of these categorizations.
- The preliminary experiments seemed to point out that the *Emotion* class (i.e. soft, neutral, aggressive) is predominantly correlated with timbre, suggesting that the improvement of timbre similarity measures could also improve classification accuracies for this categorization.

There are a number of possible explanations for these results. As already mentioned, the training data might be inconsistently labeled, as some of the categorizations are ill-defined and could also depend on the current mood of the listener. Also, although there are some features that capture temporal aspects of the pieces, most of the features only describe short-time properties of the music. As the change in time is important in music, the time-independence of most features might also play a role. Generally, the acoustic aspect is only one side of music. It is also very important how the acoustical events are combined; this is the "meaning" and semantic part of music. For this part, also socio-cultural aspects are very important, which can not be inferred from the audio signal.

Our results seem to confirm the (negative) results of [3], who mention the probable existence of a "glass ceiling", i.e. an upper bound on the performance of audio-only based similarity (or classification) algorithms. Maybe the development of dedicated features or attributes for some of the categories discussed here could contribute to the research for improved music audio features that enable us to get closer to this upper bound.

But generally, the results presented in this paper give further indication that for a holistic algorithm-based view on music, it is necessary to acquire information on the pieces from several sources: Besides audio description techniques, also extra-musical information may be important; that includes cultural aspects, usage patterns and listening habits, and - of course - the lyrics of musical pieces. There is quite some research currently going on on the automatic extraction of cultural meta-data about music from the web ([22, 24, 4, 12]), and combining this information with audio-based features in an effective way will be the next challenge.

6. ACKNOWLEDGMENTS

This research is funded by the EU FP6 project 507142 SIMAC ("Semantic Interaction with Music Audio Contents"), and by the Austrian Fonds zur Förderung der Wissenschaftlichen Forschung under grant no. Y99-START. The Austrian Research Institute for Artificial Intelligence also acknowledges the financial support of the Austrian Federal Ministries of Education, Science and Culture and of Transport, Innovation and Technology.

7. REFERENCES

[1] Iso/iec 15938 Information technology – Mpeg7 Multimedia content description interface – Part 4: Audio.

[2] J.-J. Aucouturier and F. Pachet. Music similarity measures: What's the use? In Ircam, editor, *Proceedings of the 3rd International Symposium on Music Information Retrieval*, Paris, France, October 13-17 2002.

[3] J.-J. Aucouturier and F. Pachet. Improving timbre similarity: How high is the sky? *Journal of Negative Results in Speech and Audio Sciences*, 1(1), 2004.

[4] S. Baumann, T. Pohle, and V. Shankar. Towards a socio-cultural compatibility of mir systems. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR'04)*, Barcelona, Spain, October 10-14 2004.

[5] A. Berenzweig, D. Ellis, and S. Lawrence. Using voice segments to improve artist classification of music. In *AES 22nd International Conference*, Espoo, Finland, June 15–17 2002.

[6] E. Brochu. *milq*. PhD thesis, The University of British Columbia, February 2004.

[7] S. Essid, G. Richard, and B. David. Efficient musical instrument recognition on solo performance music using basic features. In *Proceedings of the AES 25th International Conference*, London, UK, June 17-19 2004.

[8] F. Gouyon, S. Dixon, E. Pampalk, and G. Widmer. Evaluating rhythmic descriptors for musical genre classification. In *Proceedings of the AES 25th International Conference*, London, UK, June 17-19 2004.

[9] F. Gouyon, P. Herrera, and P. Cano. Pulse-dependent analysses of percussive music. In *Proceedings of the AES 22nd International Conference on Virtual, Synthetic and Entertainment Audio*, St. Petersburg, Russia, June 1-3 2002.

[10] O. Hellmuth, E. Allamanche, J. Herre, T. Kastner, N. Lefebvre, and R. Wistorf. Music genre estimation from low level audio features. In *Proceedings of the AES 25th International Conference*, London, UK, June 17-19 2004.

[11] Y. Kim and B. Whitman. Singer identification in popular music recordings using voice coding features. In *Proceedings of the International Symposium on Music Information Retrieval ISMIR-2002*, Paris, France, October 13-17 2002.

[12] P. Knees, E. Pampalk, and G. Widmer. Artist classification with web-based data. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR'04)*, Barcelona, Spain, October 10-14 2004.

[13] D. Li, I. K. Sethi, N. Dimitrova, and T. McGee. Classification of general audio data for content-based retrieval. *Pattern Recogn. Lett.*, 22(5):533–544, 2001.

[14] T. Li and M. Ogihara. Detecting emotion in music. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR'03)*, Baltimore, MD, USA, October 26-30 2003.

[15] D. Liu, L. Lu, and H.-J. Zhang. Automatic mood detection from acoustic music data. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR'03)*, Baltimore, MD, USA, October 26-30 2003.

[16] B.-S. Ong and P. Herrera. Computing structural descriptions of music through the identification of representative excerpts from audio files. In *Proceedings of the AES 25th International Conference*, London, UK, June 17-19 2004.

[17] F. Pachet and A. Zils. Automatic extraction of music descriptors from acoustic signals. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR'04)*, 2004.

[18] E. Pampalk. A matlab toolbox to compute music similarity from audio. In *Proceedings of the Fifth International Conference on Music Information Retrieval (ISMIR'04)*, Barcelona, Spain, October 10-14 2004.

[19] M. Schedl. An explorative, hierarchical user interface to structured music repositories. Master's thesis, Vienna University of Technology, Institut für Medizinische Kybernetik und Artificial Intelligence der Universität Wien, 2003.

[20] S. Streich and P. Herrera. Towards describing perceived complexity of songs: Computational methods and implementation. In *Proceedings of the AES 25th International Conference*, London, UK, June 17-19 2004.

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

[22] B. Whitman and S. Lawrence. Inferring descriptions and similarity for music from community metadata. In *Proc Intl Computer Music Conf*, 2002.

[23] B. Whitman, D. Roy, and B. Vercoe. Learning word meanings and descriptive parameter spaces from music. In *Proceedings of the HLT-NAACL03 Workshop on Learning Word Meaning from Non-Linguistic Data*, pages Edmonton, Canada, 2003.

[24] B. Whitman and P. Smaragdis. Combining musical and cultural features for intelligent style detection. In *Proc ISMIR*, 2002.

[25] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco, 2000.

[26] C. Xu, N. C. Maddage, X. Shao, and F. C. Tian. Musical genre classification using support vector machines. In *Proceedings of IEEE ICASSP03*, Hong Kong, China, April 6-10 2003.