ARCHIVES OF ACOUSTICS
33, 4, 483–496 (2008)

MUSIC INFORMATION ANALYSIS AND RETRIEVAL TECHNIQUES

Bożena KOSTEK, Łukasz KANIA

Gdańsk University of Technology
Multimedia Systems Department
Narutowicza 11/12, 80-952 Gdańsk, Poland
e-mail: bozenka@sound.eti.pg.gda.pl,
lukasz.kania@googlemail.com

(*received June 15, 2008; accepted October 15, 2008*)

This paper presents the main issues related to music information retrieval (MIR) domain. MIR is a multi-discipline area. Within this domain, there exists a variety of approaches to musical instrument recognition, musical phrase classification, melody classification (e.g. query-by-humming systems), rhythm retrieval, high-level-based music retrieval such as looking for emotions in music or differences in expressiveness, music search based on listeners' preferences, etc. The key-issue lies, however, in the parameterization of a musical event. In this paper some aspects related to MIR are shortly reviewed in the context of possible and current applications to this domain.

Keywords: music, music information retrieval, music exploration systems, multimedia databases.

1. Introduction

The aim of this study is to present key issues of Music Information Retrieval (MIR) systems [5, 12], now so very rapidly developing, that they may be recognized as a separate branch of music informatics [16]. Traditional databases store textual information about music. The searched information may be returned in the form of text or as a file containing music, and the database may be queried about performers or titles of musical compositions. As additional services, such databases offer listening to or the purchase of the composition.

Digital music databases storing musical signals in a digital form can be easily searched through. Such databases are vast Internet repositories of music information (e.g. MDL – Music Digital Libraries, Music Business Directories MBD, or distributed Multimedia Databases). They store multimedia content (e.g. about music recordings) and can be queried using non-textual criteria (e.g. through presenting a piece of recording). A simplistic, intuitive approach to multimedia content search could be a direct comparison of e.g. WAV files with recordings stored in databases. The disadvantage

of such solution is, however, a huge size of a database which inflicts problems in data searching (as binary comparison of files must be performed), dilemmas with IP rights, and generally has a very low efficiency. Thus, this approach is not a desirable one. The solution is parameterization, which can be shortly described as a process of analyzing the recording in order to remove unnecessary information [11, 14]. The process intends to retrieve only the information (parameters) that uniquely identifies a given recording. In such case, a database can store only parameters (not recordings). The recordings must also be parameterized, so that searching can be done through determining the set of parameters that best match the search criteria. The MBD databases store only metadata which are: textual information (e.g. titles, albums, year of production, music genre, etc.), parameters that describe recordings, some additional info such as IP rights, and the reference to the recording to enable replaying or buying it. Sound recordings are often stored on a separate sever, outside the searched MBD database.

The diversity of musical trends and genres, uncommon instruments as well as the variety of performers and their compositions implies the multiplicity of music recognition systems [7, 17, 27]. Nowadays systems can recognize sounds, e.g., vocal and instruments, distinguish musical phrases or audio files, and classify musical genres. The most common search tool for an Internet user is a "query-by-humming/whistling" or "query-by-example" mechanism. However, nowadays databases offer not only these traditional methods. A more and more popular network versions of MIR systems use the technology based on so-called fingerprinting. Such databases are very willingly implemented by commercial companies e.g., *Shazam*. Such services can accept an on-line stream of data that is then used to find a matching piece of music (multimedia object).

The diversity of MIR systems target applications spawns numerous methods of parameterization and classification. A common methodology in sound recognition is to examine pitch and timbre (accomplished through the detection of fundamental frequency), for this purpose multimedia databases are built up [23]. Musical phrase recognition is very often performed based on melodic and rhythmic representations of the MIDI musical material, or using metadata. In the second case Optical Mark Recognition (OMR) is now more widely and frequently used. Another popular technique is Dynamic Time Warping (DTW) which allows comparison of two music sounds independently from their rhythmic pattern and length. A separate, and very complex domain of musical information search is automatic recognition of melodic line in polyphonic audio, which addresses the problem of separation of musical sources/instruments that are acoustically or synthetically mixed [12]. Other research studies involve searching for rhythmical patterns in a melody line [25], try to recognize composer/epoch of the given music excerpt or fingerprinting of a music performer [2] or simply to recognize a class of an instrument [15].

The process of parameterization involves low-level spectral, time or wavelet parameters as well as high-level parameters of the MPEG 7 standard [4, 18]. The problem of parameterization is not limited to selection of the most significant features [12]. In searching huge databases, a key issue is to optimize the representation of the parameter

set in order to minimize the complexity of algorithmic computations. A very significant factor is also the type of the classifier used.

This study introduces some selected systems of musical information retrieval and presents an example of such experiments realized in Multimedia Systems Department.

2. Parametrization

The aim of music parameterization is to find a set of features that describe an event or a musical object [11, 13]. The process of parameterization is shown in Fig. 1. In order to make the feature extraction possible, it is necessary to properly pre-process the analyzed signal, which often means its conversion to the digital form, e.g., a 16-bit PCM mono file (5–44.1 kHz sampling rate). Then, the signal is segmented into frames of equal length and undergoes the operation of windowing. This protects against frame discontinuities and window leakage. Additionally, the overlapping of time frames may be applied. Signals prepared in such a way are ready for target calculations (pitch extraction) that extract specified features of a musical event. The final result of parameterization should be a vector of uncorrelated features that represents the musical signal. Features strongly correlated should be mapped to a single parameter to obtain one vector of features orthogonal to the musical event. Thus, after feature extraction, some post-processing is necessary to minimize the feature vector (FV) size. Post-processing results in the arrangement of provided information and in the reduction of computational complexity. To this end statistical methods are used, e.g., Fisher's statistics or methods of features correlation assessment [11].

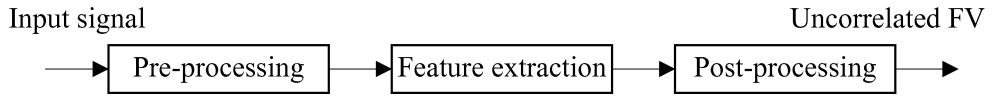

Fig. 1. Parameterization process.

The parameterization of musical sounds relates to two basic features of a sound, which are pitch and timbre. Pitch is determined through the fundamental frequency. The methods of fundamental frequency detection have a long history of development and numerous methodological solutions have already been implemented in this field [3, 20, 21].

There are three basic approaches to fundamental frequency detection. They are based on time methods, spectral methods, or hybrid methods, i.e., spectral-time methods [3]. Detection proceeds accordingly to the concept illustrated in Fig. 1. The input signal (system input) is a fragment of a musical instrument recording, while the output signal of this pre-processing block is a vector of the estimated fundamental frequency, on the basis of which a feature extraction is performed. Another elementary feature of a sound is its timbre. Timbre is a subjective feature that enables to discern two sounds of the same pitch, loudness and duration (within the same acoustic conditions). Timbre has such perceptual characteristics as brightness, richness, or sharpness. Before low-level audio TimbralSpectral and TimbralTemporal descriptors of MPEG 7 [4] standard

were formally defined and introduced, the detection of timbre was realized based on various frequency domain parameters. However, to make it really competent an appropriate composition of time and time-frequency parameters should have been used in addition.

A parameterization vector that best identifies a given musical instrument can be formed by different combinations of parameters describing the features mentioned earlier. The process of musical sound parameterization is a well-known practice [10, 13, 14], thus this study focuses on the parameters that are used to describe musical signals.

The parameterization of audio signals differs in analytical approach from the parameterization of musical instruments. This fact results from a much more complex nature of audio signals. In the case of musical signal analysis, it is necessary to examine several musical lines, a vocal part and very often the accompaniment. High complexity causes that the adaptation of such techniques as fundamental frequency detection or periodicity assessment used in simple sounds examination becomes useless. These techniques can only be helpful when testing short audio fragments, e.g., in the extraction of a solo fragment or of a single instrument. The MPEG 7 standard, or to be more precise the ISO/IEC 15938 "Multimedia Content Description Interface", is an international standard of multimedia data description developed from December 1996 to November 2001 [4]. The standard aims at normalizing the descriptors of multimedia objects (*MM objects*). It contains a set of solutions that enable to characterize a multimedia object of any form (graphics, sound, video clip, film). It defines, among other issues, the ways of data distribution, indexing, querying, and classification. The universal nature of MPEG 7 comes from the fact that the standard comprises a set of selected descriptors that represent an MM characteristics, *Description Scheme* (DS), the *Description Definition Language* (DDL) that defines descriptors and descriptor schemes, and some binary methods for descriptor coding – *Binary format forMPEG-7 data* (BiM) [10]. The DDL language is an XML Schema extension, what assures its compatibility with other standards. The functionality of ISO/IEC 15938 can be divided into the following 10 groups: Systems, Description Definition Language, Visual, Audio, Multimedia Description Schemes, Reference Software, Conformance, Extraction and use of descriptors, Profiles, Schema Definition.

The elements of the MPEG 7 are shortly presented, below. They are descriptors of melody, timbre and audio signal rhythm. Apart from metadata, which are most useful to archive music, two other types of descriptors [18, 22] are used to describe an audio signal; they are *low-level* and *high-level descriptors*. It is worth notifying that the ISO/IEC 15938 standard gives only definitions of parameters that could describe the content of a musical signal without specifying algorithms used to calculate their values. Low-level parameters are the basis for high-level parameters whose form is dependant on the application and on the musical feature described.

The MPEG-7 standard comprises 17 low-level parameters separated into 6 groups. An additional, however a very useful one, is the parameter that defines *silence*. The MPEG-7 standard defines high-level descriptors of audio signals as well. High-level descriptors specify the way low-level parameters are used in specific applications. The description schemes are divided into five groups [4]:

1. Audio Signature Description Scheme – a group of low-level descriptors that uniquely describe an audio signal. Parameters of this group are used in systems based on so-called *fingerprints*.

2. Musical Instrument Timbre Description Tools – descriptors from this group describe perceptual features of musical instrument sounds, and they relate to timbre. The MPEG 7 standard represents low-level descriptors for four classes of musical sounds: harmonic, coherent, sustained sounds, and non-sustained, percussive sounds, and they are specifically recommended for two classes such as: sustained harmonic sounds and non-sustained percussion sounds. Notions such as "attack", "brightness" or "richness" of a sound belong to this group.

3. Melody Description Tools – descriptors from this group aim at the recognition of melodic information in monophonic recordings. They can be successfully applied in *Query-by-Humming* systems to search for compositions that match a given melodic line.

- MelodyContour Description Scheme – a scheme that defines melody contour as a 5-step contour in which a pitch of a given note is represented with regard to the previous note (representing the interval difference between adjacent notes). Additionally, information about rhythm is stored as the number of the nearest whole beat of each note.
- MelodySequence Description Scheme – a set of descriptors used to retrieve more precise information about a melody than the MelodyContour DS can provide. To achieve this, a precise difference between pitches of a current and a previous note is written down (e.g., in cents). It is stored along with the information on the rhythm obtained through coding the logarithm of the ratio between the start times of adjacent notes. Optionally, depending on the application, a series of support descriptors such as lyrics, key, meter, and starting note, may be used.

4. General Sound Recognition and Indexing Description Tools – a group of descriptors that serve to automatically index, segment, categorize, and recognize sounds:

- SoundModel Description Scheme – a model that uses low-level Spectral Basis group descriptors. A series of states that comprise a statistical model is worked out based on low-level parameters, most commonly hidden Markov or Gauss mixture model is used to achieve that.
- SoundClassificationModel Description Scheme – a scheme that uses the set of *SoundModel* models to create a multi-classifier that categorizes segments of audio signals according to conditions defined by the classification scheme. This results in recognition of sounds classes, such as speech and music, even narrower categories such as male, female, or particular instrument class. Other possible applications include genre classification and voice recognition.
- SoundModelStatePath – a descriptor that collects series of indexes indicating states of a SoundModel model for a given sound. The descriptor is a concise description of a sound segment, and along with other models, it may be used for fast comparison of other models.

- SoundModelStateHistogram – a descriptor containing a normalized histogram of *SoundModel* states for a given segment of a sound. State histograms are used to compare audio signal segments.

5. Spoken Content Description Tools – a group formed from parameters related to speech recognition, and defined by two classes:

- SpokenContentLattice Description Scheme – a class containing indexes of nodes connected by words or phonemes, in which each connection indicates a word or phoneme defined in the vocabulary of the automatic speech recognition system. The nodes are defined by their location on the time axis (timeOffset) with regard to the lattice structure. This structure allows defining the character of pronunciation through the use of various phonetic elements combinations between words and phonemes.
- Spoken Content Header – a class that holds the number of components used by *SpokenContentLattice DS*. It contains two basic descriptors: *WordLexicon* and *PhoneLexicon*. The first descriptor is an indexed list of characters describing the set of words of a recognizer. The second, contains the lattice list that is the resource of phonetic components of the automatic speech recognition system. Optionally, *ConfusionInfo* and *SpeakerInfo* descriptors are used. They define the linguistic error matrix for each position of *PhoneLexicon* and such biometric characteristics of the speaker as his/her vocabulary, language, speaking habits and personal data.

Concise comparison of audio features extraction tools is presented in Table 1 [17]. As stated by Lartillot, this table shows existing classification frameworks recognizable in the Internet.

Table 1. Comparison of feature extraction tools [17].

System	Features	Interface	Output	Batch process	Dependency control	Distributed computing	Memory management
Marsyas	dozen low + beats	scripting language	export	yes	manual	yes	real time
jAudio	$\sim$ 20 low + beats	GUI	export	yes	auto	yes	yes
CLAM	$\sim$ 12 spectr. + tonal	visual program.	pre-visual. & export	yes	manual	?	real time
ChucK	dozen low-level	program. language	numerical	yes	manual	yes	real time
M_2K	dozen low-level	D2K visual program.	graphic & export	yes	manual	yes	yes
Psysound	$\sim$ 30 low & high	GUI	graphic & export	yes			
IPEM toolbox	17 low & high	Matlab functions	graphic & numerical				
MA toolbox	8	Matlab functions	numerical values				
MIRtool-box	$\sim$ 40 low & high	adaptive syntax	graphic & export	yes	manual	Future using DC tb	yes

3. Selected algorithms and tools for music classification

Automatic musical sound classification, similarly as many other categorizations, is based on specifying several groups described by characteristic descriptors (objective or subjective ones). However, classification has special meaning in automatic recognition of music. Recognized objects may be: instruments, genres, particular compositions, or sounds. The classification criteria may, depending on the type of recognized objects, be timbre of the instrument, rhythm, pace, or spectral content characterized by appropriate parameters. The course of classification, however, does not depend upon criteria or classified objects, and it is in most cases identical for a given algorithm. Having extracted an uncorrelated vector of features in the process of parameterization, it is possible to classify sounds using any algorithm. The most simple are such classifiers as minimum-distance methods based on various metrics, discriminant analysis or Bayes classifiers. Advanced methods of classification are neural networks, decision trees, genetic algorithms [8], SVM – Support Vector Machine [1] (and a special case of its implementation, called: Sequential Minimal Optimization (SMO) [6], Hidden Markov Model or Rough Sets [19]. Support Vector Machine is a supervised learning algorithm developed over the past decade by VAPNIK [26] and later by others. The SVM algorithm performes by mapping the given training set into a possibly high-dimensional feature space and attempting to locate in that space a plane that separates the positive from the negative examples. Having found such a plane, the SVM can then predict the classification of an unlabeled example by mapping it into the feature space and asking on which side of the separating plane the example lies [9]. SMO algorithm is Platt's sequential minimal optimization algorithm for training a support vector classifier. This implementation globally replaces all missing values and transforms nominal attributes into binary ones. The Rough Set-based methods do not require a vector of uncorrelated features. Due to the fact that most of these methods are frequently used to solve different problems in acoustics, they have only been mentioned here [16]. However, a Reader may found implementation of such algorithms in WEKA classification system [7].

4. Experiments

One of the problems being solved in Multimedia Systems Department of the Gdańsk University of Technology is the classification of music genres. The example of such classification was illustrated with selected algorithms from WEKA system [7]. For this purpose a database of complete pieces of music (in mp3 format) was created. The files were collected from two Internet portals that offer free music, i.e., http://www.mp3.wp.pl/strefa and http://download.music.com. The collection comprises 15 styles of music *blues, britpop, classical orchestral, country*, *dance, folk, grunge, hip-hop, jazz, metal, new age, punk, R&B (rhythm-and-blues*), *roots reggae, techno*. Each style is represented by 16 pieces. The database holds 240 pieces of music in total. They belong to various artists and occupy 1.2 GB of disk storage.

The experiments began with feature extraction tasks, i.e. the feature vector was defined and examined to specify which of the features are adequate to classify music styles. As the MPEG 7 parameters are now broadly used in research on classification of musical objects, their definitions will not be mentioned here. Table 2 presents the list of parameters contained in the feature vector and their interpretation. The next step was to verify the feature vector with Fisher's statistics (F) [11]. The concept of

Table 2. Content of the feature vector.

No.	Parameter	Meaning
1	AudioPower mean (Apm)	average loudness of fragment
2	AudioPower std dev. (Apstdev)	describes the temporally-smoothed instantaneous power, dynamics, parameter does not depend on tempo changes
3–36	AudioSpectrumEnvelope (ASE)	describes the short-term power spectrum of an audio signal and timbre, analysis in 1/3 octave bands
37–60	AudioSpectrumFlatness (ASF)	describes the flatness properties of the spectrum of an audio signal for each of a number of frequency bands. When this vector indicates a high deviation from a flat spectral shape for a given band, it may signal the presence of tonal components, analysis in 1/3 octave bands for timbre differentiating
61	AudioSpectrumCentroid (ASC)	an analogue to the SpectralCentroid, but defined on a logarithmic frequency scale, helps distinguishing between pure-tone and noise-like sounds, differentiating vocal and instrumental pieces
62	AudioSpectrumSpread (ASS)	indicates whether the power spectrum is centred near the spectral centroid, or spread out over the spectrum variety of instruments and vocals, useful for differentiating variety of instruments and vocals
63	Spectral Centroid (SC)	timbre of spectrum correlated with sound sharpness
64	SpectralFlux (SF)	a measure of short-time changes of the spectrum; separation of speech from music
65	TemporalCentroid (TC)	represents gravity center of time envelope, characterizes the signal envelope, representing where in time the energy of a signal is focused. It may distinguish between a decaying piano note and a sustained instrument note, when the lengths and the attacks of the two notes are identical.
66	ZeroCrossingRate (ZCR)	represents the rate at which zero crossings occur; a simple measure of the frequency content of a signal, also allows for estimating noisiness of piece
67, 68	RollOff85 (Roll85), RollOff95	measure of spectral shape; frequency for which the energy of spectrum reaches 85%/95% in total

F statistics is to examine in pairs each representative of a class with every parameter. The higher the absolute value $|V|$ of the statistics is, the easier becomes separation of the two classes using the tested parameter. F statistics is defined by the following formula [11]:

$$V = \frac{\overline{X} - \overline{Y}}{\sqrt{S_1^2/n + S_2^2/m}}, \tag{1}$$

where

$\overline{X}$, $\overline{Y}$ are mean parameter values:

$$\overline{X} = \frac{1}{n} \cdot \sum_{i=1}^{n} X_i, \qquad \overline{Y} = \frac{1}{m} \cdot \sum_{i=1}^{m} Y_i; \tag{2}$$

S_1^2, S_2^2 are variance estimators of respective random variables:

$$S_1^2 = \frac{1}{n-1} \cdot \sum_{i=1}^{n} (X_i - \overline{X})^2, \qquad S_2^2 = \frac{1}{m-1} \cdot \sum_{i=1}^{m} (Y_i - \overline{Y})^2 \tag{3}$$

and n, m are cardinality of populations X and Y accordingly.

The analysis of F statistics enabled to determine the best and the worst pairs of genres in terms of their recognition. F statistics also indicated the most discriminatory parameters of the vector. The example of the analysis results (the pairs easiest to be recognized with respect to maximum Fisher values) for samples of 3 seconds are shown in Table 3.

Table 3. Best discernible pairs (in terms of maximum F values for 3 s samples).

Genre pairs	F value	Par.	F value	Par.	Fvalue	Par.	F value	Par.	F value	Par.
Classical – punk	28.04	ASF16	24.75	ASF15	23.42	ASF14	22.80	ASF19	21.80	ASF20
classical – hip-hop	26.32	Roll95	23.19	Roll85	21.25	SC	18.79	ASF16	18.71	ASF20
classical – reggae	24.51	Roll95	24.00	Roll85	20.76	SC	16.15	ASF20	14.81	ASF16
classical – grunge	24.51	Roll95	23.03	Roll85	20.16	SC	18.78	ASF16	17.73	ASF15
classical – dance	22.04	Roll95	18.44	Roll85	16.83	ASE33	16.56	ASF16	16.51	ASF14
classical – metal	23.90	ASF16	23.08	ASF15	21.55	ASF14	20.30	Roll85	20.06	ASF20
classical – *R&B*	23.14	Roll95	21.71	Roll85	18.54	SC	16.96	ASE33	16.45	ASF20
classical – techno	22.52	ASF16	22.44	ASF20	21.21	ASF15	21.21	ASF19	20.05	ASE34
country – punk	21.64	ASF14	21.34	ASF16	21.07	ASF12	20.10	ASF11	20.05	ASF18
punk – reggae	21.36	ASF16	19.75	ASF18	19.06	ASF17	17.14	ASF19	16.81	Apm
punk – *R&B*	20.87	Apm	17.46	ASF18	16.45	ASE25	16.41	ASF16	15.21	ASF19
blues – punk	20.39	ASF16	16.17	ASF11	15.91	ASF18	15.29	ASF15	15.23	ASF14

Definitely, the best discriminators of music styles are the AudioSpectrumFlatness (ASF), SpectralCentroid (SC), and the RollOff descriptors. The parameters that turned

to be of no value for classification of musical genres in this study are Spectral Flux (SF) and TemporalCentroid (TC). AudioSpectrumCentroid (ASC) and AudioSpectrumSpread (ASS) also showed little significance. The easiest to tell apart were classic and punk music. It was rather difficult to differentiate between *R&B*, *blues*, *jazz* and *reggae*.

Classification of genres was carried out on 2400 three-second samples (10 fragments of each piece making 160 pieces of the same genre in total). Three classifiers from WEKA were used: NN (*Nearest Neighbor*), k-NN (k-*Nearest Neighbor*), and SMO PUK (Sequential Minimal Optimization, PUK *kernel*). For supporting vectors (SVM) algorithms, the most important optimization decision is to properly choose the kernel function. From several kernel functions implemented in the WEKA system, the best results were obtained for the PUK kernel, the so-called Pearson VII function-based universal kernel [6].

The classification was performed within the 10-fold cross-validation scheme. The set of data was partitioned into 10 subsamples. Of these 10 subsamples, one subsample was retained as the validation data, and the remaining 9 subsamples were used as training data. The set of 2400 samples, each having duration time of 3 seconds, and a non-reduced vector of the following features: [Apm, Apstdev, ASE1,..., ASE34, ASF1,..., ASF24, ASC, ASS, SC, SF, TC, ZCR, RollOff85, RollOff95] was employed in classification tests. Test results are presented in Tables 4–6. In the case of the k-nearest

Table 4. Classification efficiency [%] for the NN algorithm based on various metrics.

Music genre	Euclidean metric [%]	Manhattan metric [%]	Chebyshev metric [%]
Blues	78.1	83.8	68.1
Britpop	77.5	80.6	55.6
Classical	67.5	71.9	53.1
Country	74.4	77.5	60
Dance	73.8	75	58.1
Folk	91.3	87.5	76.9
Grunge	86.9	88.1	62.5
Hip-hop	88.1	88.1	76.9
Jazz	65.6	70.6	46.9
Metal	88.8	91.3	75
New age	78.1	79.4	57.5
Punk	95	96.3	85.6
Reggae	70.6	73.8	58.1
R&B	71.9	75.6	51.3
Techno	85	84.4	69.4
Overall	79.5	81.58	63.67

Table 5. Classification efficiency [%] for the k-NN algorithm based on metrics and the type of distance weighting for $k = 3$.

Weighting	Euclidean metric [%]	Manhattan metric [%]	Chebyshev metric [%]
NO weighting	73.917	76.542	56.208
1/dist.	78.708	80.708	62.542
1-dist.	78.417	80.693	62.25

Table 6. Classification results [%] for SMO PUK algorithm.

Music genre	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**m**	**n**	**o**	Classif. Eff. [%]
Blues (**a**)	**106**	13	7	7	1	5	2	2	3	0	1	0	1	2	1	66.3
Britpop (**b**)	7	**108**	1	6	3	2	10	0	9	4	0	3	2	3	2	67.5
Classical (**c**)	5	2	**135**	7	0	2	0	0	2	0	6	1	0	0	0	84.4
Country (**d**)	5	1	8	**114**	1	9	2	4	3	0	3	0	3	7	0	71.3
Dance (**e**)	4	2	3	6	**111**	2	1	4	2	0	3	0	5	6	11	69.4
Folk (**f**)	6	3	4	3	1	**128**	1	2	4	0	0	0	3	4	1	80
Grunge (**g**)	3	1	1	4	1	1	**135**	1	3	5	1	0	0	3	1	84.4
Hip-hop (**h**)	4	1	0	2	2	0	1	**137**	1	0	0	0	3	8	1	85.6
Jazz (**i**)	1	0	0	13	12	3	9	0	**197**	0	6	0	3	3	3	60.6
Metal (**j**)	0	2	7	0	0	0	1	0	1	**137**	5	7	0	0	0	85.6
New age (**k**)	1	0	16	6	3	2	0	1	5	0	**115**	0	4	0	7	71.9
Punk (**l**)	2	0	0	0	0	1	4	0	0	10	0	**143**	0	0	0	89.4
R&B (**m**)	11	0	0	9	2	0	0	17	3	0	1	0	**105**	10	2	65.6
Reggae (**n**)	11	0	1	9	2	2	1	9	4	0	0	0	16	**100**	5	62.5
Techno (**o**)	0	1	4	0	11	0	2	2	3	0	2	0	0	0	**135**	84.4

neighbor algorithm, the best results were achieved for the Manhattan metric, known also under the notion of taxicab metric. The optimum configuration of the k-NN classifier is obtained when vectors are weighted with the coefficient inversely proportional to the distance to three neighbors (assuming the taxicab metric is used). Among the three discussed classifiers chosen from the WEKA system, the distance-based methods produce the best results in terms of efficiency and the time of calculations.

Diagonal of Table 6 (bold face numbers) shows the number of objects belonging to the given class, recognized correctly.

Undoubtedly, in order to best recognize music genres, the feature vector should be supplemented with additional parameters. Having in mind that the objective is to increase the efficiency of parameterization, several dozens of musical parameters offered

by the *MIRToolbox* of MATLAB system were explored. Fisher's statistics – calculated over a distinctive set of data comprising 240 musical fragments each lasting 3 second – allowed initial selection of candidate features that would constitute an appropriate supplement to the feature vector. Among others, these were statistical values of certain mel-cepstral parameters and the distribution of sound pitches.

The examination of both Fisher's statistics values and the results of classification show that, notwithstanding the size of the database, some styles are more difficult to recognize than others. A solution to this problem seemed to be mapping these styles to the same label. Thus, the experiments were repeated, with the classes of *britpop*, *blues* and *grunge* labeled *rock*, the classes of *dance* and *techno* labeled *electronica*. Labels of genres that caused greatest problems in recognition (i.e., *jazz*, *R&B*, *punk*, *folk*, *reggae* and *country*) were eliminated. In the context of discussed experiments, the best results were achieved for distance-based classifiers and support vectors used with polynomial kernel functions.

5. Conclusions

The study reviews problems of the classification and search of musical objects. Additionally, it presents a series of experiments with different methods of classification. The designed feature vector was tested in the WEKA system over the set of 2400 sound samples each 3 seconds long (1600 fragments of every genre). Selected classifiers – NN, k-NN, support vectors method (SMO) – were optimized using presented closed set of samples. The results of the research indicated that parameters *TC* and *SF* should be rejected. Classification carried out with the reduced vector resulted in 80% efficiency of recognition for distance-based methods and in 70% efficiency for *TC* and *SF*.

The process of automatic classification proved the complexity of recognition tests in real conditions. Efficiency of the classifiers (for tests operating over distinctive fragments) that oscillated around 60–80% confirmed the implemented parameters to be appropriate for recognition types such as *Query-by-Example*. Genre recognition, however, required class labels to be reorganized or supplemented with additional parameters. Finally, an overall efficiency of 70% was achieved for musical style recognition with the SMO algorithm for *rock*, *classical*, *metal*, *new age*, *electronica* and *hip-hop* types.

References

[1] BURGES C.J.C., *A Tutorial on Support Vector Machines for Pattern Recognition*, Kluwer Academic Publisher, Boston 1998.

[2] CHUDY M., *Automatic identification of music performer using the linear prediction cepstral coefficients method*, Archives of Acoustics, **33**, 1, 27–33 (2008).

[3] DZIUBINSKI M., KOSTEK B., *Octave Error Immune and Instantaneous Pitch Detection Algorithm*, J. New Music Research, **34**, 273–292 (2005).

[4] http://www.chiariglione.org/MPEG/standards/mpeg-7/mpeg-7.htm#E12E46 (information on MPEG 7 standard).

[5] http://ismir2007.ismir.net (International Conference on Music Information Retrieval website).

[6] http://weka.classifiers.functions.supportVector (information on SVM implemented in WEKA system).

[7] http://www.cs.waikato.ac.nz/ml/weka/ (information on WEKA classifier package).

[8] http://www.soft-computing.de/def.html (information on soft computing methods).

[9] http://svm.sdsc.edu/svm-overview.html (information on SVM).

[10] KIM H.-G., BURRED J., SIKORA T., *How efficient is MPEG-7 for general sound recognition*, Proceedings of the Audio Engineering Society (AES04) International Conference, London. Retrieved, March 16, 2004.

[11] KOSTEK B., *Soft Computing in Acoustics, Applications of Neural Networks, Fuzzy Logic and Rough Sets to Musical Acoustics, Studies in Fuzziness and Soft Computing*, Physica Verlag, Heildelberg, New York 1999.

[12] KOSTEK B., *Perception-Based Data Processing in Acoustics. Applications to Music Information Retrieval and Psychophysiology of Hearing*, Springer Verlag, Series on Cognitive Technologies, Berlin, Heidelberg, New York 2005.

[13] KOSTEK B., CZYŻEWSKI A., *Representing Musical Instrument Sounds for their Automatic Classification*, J. Audio Eng. Soc., **49**, 768–785 (2001).

[14] KOSTEK B., WIECZORKOWSKA A., *Parametric representation of musical sounds*, Archives of Acoustics, **22**, 1, 3–26 (1997).

[15] KOSTEK B., KRÓLIKOWSKI R., *Application of artificial neural networks to the recognition of musical sounds*, Archives of Acoustics, **22**, 1, 27–50 (1997).

[16] KOSTEK B., *Applying computational intelligence to musical acoustics*, Archives of Acoustics, **32**, 3, 617–629 (2007).

[17] LARTILLOT O., *MIR toolbox 1.0 User Guide*, University of Jyvaskyla, Finland 2007.

[18] LINDSAY A., HERRE J., *MPEG-7 and MPEG-7 Audio – An Overview*, **49**, 7–8, 589–594 (2001).

[19] PAWLAK Z., SKOWRON A., *Rough sets and Boolean reasoning*, Information Sciences, **177**, 1, 41–73 (2007).

[20] RABINER L., *On the use of autocorrelation analysis for pitch detection*, IEEE Trans. ASSP, **25**, 24–33 (1977).

[21] SZCZERBA M., CZYŻEWSKI A., *Pitch Detection Enhancement Employing Music Prediction*, J. Intelligent Information Systems, **24**, 2–3, 223–251 (2005).

[22] WUST O., CELM O., *An MPEG-7 Database System and Application for Content-Based Management and Retrieval of Music*, ISMIR 2004, Barcelona, Spain, October 10–14, 2004.

[23] WITULSKI B., ŁUKASIK E., *Multimedia presentation of musical instruments*, Archives of Acoustics, **33**, 1, 35–41 (2008).

[24] WOŁKOWICZ J., KULKA Z., KEŠELJ V., *n-gram-based approach to composer recognition*, Archives of Acoustics, **33**, 1, 43–55 (2008).

[25] WÓJCIK J., KOSTEK B., *Computational complexity of the algorithm creating hypermetric rhythmic hypotheses*, Archives of Acoustics, **33**, 1, 57–63 (2008).

[26] VAPNIK V.N., *Statistical Learning Theory*, Wiley, New York 1998.

[27] ZWAN P., KOSTEK B., *System for Automatic Singing Voice Recognition*, J. Audio Eng. Soc., **56**, 9, 710–723 (2008).