Application of temporal descriptors to musical instrument sound recognition

Alicja A. Wieczorkowska [1] (alicja@pjwstk.edu.pl)
Jakub Wróblewski [1] (jakubw@pjwstk.edu.pl)
Piotr Synak [1] (synak@pjwstk.edu.pl)
Dominik Ślęzak [2,1] (slezak@pjwstk.edu.pl)

1. *Polish-Japanese Institute of Information Technology ul. Koszykowa 2, 02-008 Warsaw, Poland*
2. *Department of Computer Science, University of Regina Regina, SK, S4S 0A2, Canada*

Abstract. An automatic content extraction from multimedia files is recently being extensively explored. However, an automatic content description of musical sounds has not been broadly investigated and still needs an intensive research. In this paper, we investigate how to optimize sound representation in terms of musical instrument recognition purposes. We propose to trace trends in the evolution of values of MPEG-7 descriptors in time, as well as their combinations. Described process is a typical example of KDD application, consisting of data preparation, feature extraction and decision model construction. Discussion of efficiency of applied classifiers illustrates capabilities of possible progress in the optimization of sound representation. We believe that further research in this area would provide background for an automatic multimedia content description.

Keywords: knowledge discovery in databases, music content processing, multimedia content description, MPEG-7

1. Introduction

An automatic extraction of information from multimedia databases is recently of great interest. Multimedia data available for end users are usually labeled with some information (e.g. title, time, author, etc.), but in most cases it is insufficient for content-based searching. For instance, it is not possible to automatically find in an audio CD all segments with a given tune played, e.g., by the flute. To address the task of automatic content-based search, several descriptors need to be assigned at various levels to segments of multimedia files. Moving Picture Experts Group works on MPEG-7 standard, named "Multimedia Content Description Interface" (ISO/IEC, 2002), which defines a universal mechanism for exchanging the descriptors. However, neither feature (descriptor) extraction nor searching algorithms are encompassed in


© 2003 *Kluwer Academic Publishers. Printed in the Netherlands.*

MPEG-7. Therefore, an automatic extraction of multimedia content, including musical information, should be a subject of study.

All descriptors used so far (Ando and Yamaguchi, 1993; Brown, 1999; Kostek and Wieczorkowska, 1997; Martin and Kim, 1998), reflect specific features of sound, like spectrum, time envelope, etc. We propose a different approach: we suggest an analysis of feature changes in time and taking as additional descriptors some patterns in trends observed for particular features. We discuss how to achieve this goal by applying data preprocessing and mining tools developed within the theory of rough sets introduced in (Pawlak, 1991).

The analyzed database origins from MUMS audio CD's (Opolko and Wapnick, 1987). These CD's contain sounds of musical instruments, played with various articulation techniques. We processed samples representing brass, string, and woodwind instruments of contemporary orchestra. The obtained database was divided into 18 classes, representing single instruments and selected articulation techniques.

2. Sound descriptors

2.1. MPEG-7 descriptors

Descriptors of musical instruments should allow to recognize instruments independently on pitch and articulation. Sound features included in MPEG-7 Audio are based on the research performed so far in this area and they comprise technologies for musical instrument timbre description, audio signature description and sound description. The audio description framework in MPEG-7 includes 17 temporal and spectral descriptors divided into the following groups (ISO/IEC, 2002):

1. basic: instantaneous waveform and power values
2. basic spectral: log-frequency power spectrum envelopes, spectral centroid, spectrum spread, and spectrum flatness
3. signal parameters: fundamental frequency and harmonicity of signal
4. timbral temporal: log attack time and temporal centroid
5. timbral spectral: spectral centroid, harmonic spectral centroid, and harmonic spectral deviation, harmonic spectral spread, harmonic spectral variation
6. spectral basis representations: spectrum basis and spectrum projection

Each of these features can be used to describe a segment with a summary value that applies to the entire segment or with a series of sampled values. An exception is the timbral temporal group, as its values apply only to segments as a whole.

2.2. Other descriptors

All the descriptors included in MPEG-7 are based on published research. Their number included in the standard has been limited to 17, in order to obtain compact representation of audio content for search purposes and other applications. Apart from the features included in MPEG-7 (Peeters et al., 2000), the following descriptors have been used in the research, worldwide:

- duration of the attack, quasi-steady state and ending transient of the sound in proportion to the total time (Kostek and Wieczorkowska, 1997)
- moments of the time wave (Brown et al., 2001)
- pitch variance-vibrato (Martin and Kim, 1998; Wieczorkowska, 1999b)
- contents of the selected groups of harmonics in spectrum (Kostek and Wieczorkowska, 1997), like even/odd harmonics Ev/Od

$$Ev = \frac{\sqrt{\sum_{k=1}^{M} A_{2k}^2}}{\sqrt{\sum_{n=1}^{N} A_n^2}} \qquad Od = \frac{\sqrt{\sum_{k=2}^{L} A_{2k-1}^2}}{\sqrt{\sum_{n=1}^{N} A_n^2}} \tag{1}$$

  and lower/middle/higher harmonics $Tr_1/Tr_2/Tr_3$ (Tristimulus parameters (Pollard and Jansson, 1982), used in various versions)

$$Tr_1 = \frac{A_1^2}{\sum_{n=1}^{N} A_n^2} \quad Tr_2 = \frac{\sum_{n=2,3,4} A_n^2}{\sum_{n=1}^{N} A_n^2} \quad Tr_3 = \frac{\sum_{n=5}^{N} A_n^2}{\sum_{n=1}^{N} A_n^2} \tag{2}$$

  where A_n denotes the amplitude of the n^{th} harmonic, N is the number of harmonics available in spectrum, $M = \lfloor N/2 \rfloor$ and $L = \lfloor N/2 + 1 \rfloor$
- statistical properties of sound spectrum, including average amplitude and frequency deviations, average spectrum, standard deviation, autocorrelation and cross-correlation functions (Ando and Yamaguchi, 1993; Brown et al., 2001)

– various properties of the spectrum, including higher order moments, such as skewness and kurtosis, as well as brightness and spectral irregularity (Fujinaga and McMillan, 2000; Wieczorkowska, 1999a), defined as below:

$$Br = \frac{\sum_{n=1}^{N} n\,A_n}{\sum_{n=1}^{N} A_n} \quad Ir = \log\left(20 \sum_{k=2}^{N-1} \left|\log \frac{A_k}{\sqrt[3]{A_{k-1}A_kA_{k+1}}}\right|\right) \quad (3)$$

– constant-Q coefficients (Brown, 1999; Kaminskyj, 2000)

– cepstral and mel-cepstrum coefficients and derivatives (Brown, 1999; Batlle and Cano, 2000; Eronen and Klapuri, 2000)

– multidimensional scaling analysis trajectories (Kaminskyj, 2000)

– descriptors based on wavelets (Wieczorkowska, 1999a; Kostek and Czyzewski, 2001), Bark scale bands (Eronen and Klapuri, 2000) and other (Herrera et al., 2000; Wieczorkowska and Raś, 2001)

2.3. Descriptors used in our research

The main goal of this paper is to verify, how much one can gain by analyzing widely used descriptors by means of the dynamics of their behavior in time. We restrict ourselves to a small part of the known descriptors, to be able to compare the results obtained with and without analysis of temporal behavior more clearly. We begin the analysis process with the following descriptors.

Temporal descriptors:

– *Length*: Signal length

– *Attack*, *Steady* and *Decay*: Relative length of the attack (till reaching 75% of maximal amplitude), quasi-steady (after the end of attack, till the final fall under 75% of maximal amplitude) and decay time (the rest of the signal), respectively

– *Maximum*: Moment of reaching maximal amplitude

Spectral descriptors:

– *EvenHarm* and *OddHarm*: Harmonics defined by (1)

– $Tristimulus1, 2, 3$: Tristimulus parameters given by (2)

– *Brightness* and *Irregularity*: properties defined by (3)

– *Frequency*: Fundamental frequency

The spectral descriptors were used so far in the literature only in purpose of reflecting specific features of the whole sound or in the selected time moments. In the foregoing sections, we propose to consider the same features but calculated over the chains of reasonably small time intervals. That allows to observe patterns of changes of sound descriptors in time, what is especially interesting for the attack time.

3. Musical instrument sound recognition

3.1. Classification models

One of the main goals of data analysis is to construct models, which properly classify objects to some predefined classes. Reasoning with data can be stated as a classification problem, concerning prediction of decision class basing on information provided by some attributes. For this purpose, one stores data in so called decision tables, where each training case drops into one of decision classes.

A decision table takes the form of $\mathbf{A} = (U, A \cup \{d\})$, where each attribute $a \in A \cup \{d\}$ is identified with a function $a : U \to V_a$ from the universe of objects U into the set V_a of all possible values of a. Values $v_d \in V_d$ correspond to mutually disjoint decision classes of objects. In case of the analysis of the musical instrument sound data taken from (Opolko and Wapnick, 1987), we deal with a decision table consisting of 667 objects corresponding to samples of musical recordings. We have 18 decision classes corresponding to various musical instruments – flute, oboe, clarinet, violin, viola, cello, double bass, trumpet, trombone, French horn, tuba – and their articulation – vibrato, pizzicato, and muted (Wieczorkowska, 1999b).

Methods for construction of classifiers can be also regarded as tools for data generalization. They include rule-based classifiers, decision trees, k-nearest neighbor classifiers, neural nets, etc. Problem of musical instrument sound recognition has been approached in several research studies, applying various methods. The most common one is k-nearest neighbor algorithm, applied in (Martin and Kim, 1998; Fujinaga and McMillan, 2000; Eronen and Klapuri, 2000; Kaminskyj, 2000). To obtain better results, Fujinaga and MacMillan (2000) applied k-nearest neighbor classifier to the weighted feature vectors and a genetic algorithm to set the optimal set of weights. Brown in her research (Brown, 1999) applied clustering and Bayes decision rules, using k-means algorithm to calculate clusters, and forming Gaussian probability density functions from the mean and variance of each of the clusters. Martin

and Kim (1998) used maximum a posteriori classifiers, based on Gaussian models obtained through Fisher multiple-discriminant analysis. Gaussian classifier was also used by Eronen and Klapuri (2000).

Apart from statistical methods, machine learning tools have been also applied. For instance, classification based on binary trees was used in (Wieczorkowska, 1999a). Another popular approach to musical instrument sound classification is based on various neural network techniques (Cosi et al., 1994; Toiviainen, 1996; Wieczorkowska, 1999a). Research based on hidden Markov models was reported in (Batlle and Cano, 2000), whereas Wieczorkowska (1999b) applied rough set approach to musical sound classification. Extensive review of classification methods applied to this research, including the above mentioned and other (for instance, support vector machines) is given in (Herrera et al., 2000). We go back to these issues in Section 7, while discussing the results of classification experiments.

3.2. KDD framework

All the above approaches are based on adapting the well known classifier construction methods to the specific domain of musical instrument sounds. However, the process of analyzing data is not restricted just to the classifier construction. In case of the musical instrument sound analysis, one has to extract a decision table itself – to choose the most appropriate set A of attributes-descriptors, as well as to calculate values $a(u) \in V_a$, $a \in A$, for particular objects-samples $u \in U$. Thus, it is better to write about this task in terms of a broader methodology.

Knowledge Discovery in Databases (KDD) is a process, which consists of the following steps (Düntsch et al., 2000):

- understanding application domain
- determining goal
- creating/selecting target data set
- preprocessing
- data reduction and transformation
- selection of data mining method, algorithms, and parameters
- model construction (data mining)
- interpretation of results

In case of classification of musical instruments, the first two steps comprise of the musical domain analysis. Next, the proper selection (Liu and Motoda, 1998) and reduction (Pawlak, 1991) of the set of features is crucial for efficiency of classification algorithm. In some cases, a set of attributes is worth transforming into more suitable form before it is used to model the data. For instance, before describing the data set by decision rules, one may transform attribute values to gain higher support of rules, keeping their accuracy, and increasing generality of a model. The need of such a transformation is shown for various kinds of feature domains: numeric, symbolic, as well as for time series (Ślęzak and Wróblewski, 1999; Nguyen, 2000; Synak, 2000; Wróblewski, 2000).

3.3. Extraction of temporal features

The above mentioned feature transformation is of crucial importance while considering musical instruments. Because of the nature of the musical sounds, methods concerned with time series analysis seem to be of a special interest. It is difficult to find a numerical description of musical instrument sounds that allows correct classification of instrument for sound of various pitch and/or articulation. Listener needs transients (especially the beginning of sound) to correctly classify musical instrument sounds, but during transients the sound features change dramatically and they usually differ from the sound features for the (quasi-)steady state. It is illustrated by Figure 1, where fragments of time domain representation of oboe sound a^1 of frequency 440Hz are shown.

As we can observe, the beginning (attack) of the sound significantly differs from the quasi-steady state. During the attack, changes are very rapid, but in the quasi-steady state some changes may happen as well, especially when the sound is vibrated. Feature vectors used so far in the research reflect mainly quasi-steady state and the attack of sounds. The used features are based on time domain analysis, spectral analysis, and some other approaches (for example, wavelet analysis). The time domain analysis can describe basic features applicable to any sounds, like basic descriptors from MPEG-7, or features specific for the whole sound, like timbral temporal features from MPEG-7 (see Section 2).

4. Preprocessing of musical sound data

4.1. Data description

In purpose of learning classifiers for the musical instrument sound recognition, we need to prepare the training data in the form of decision

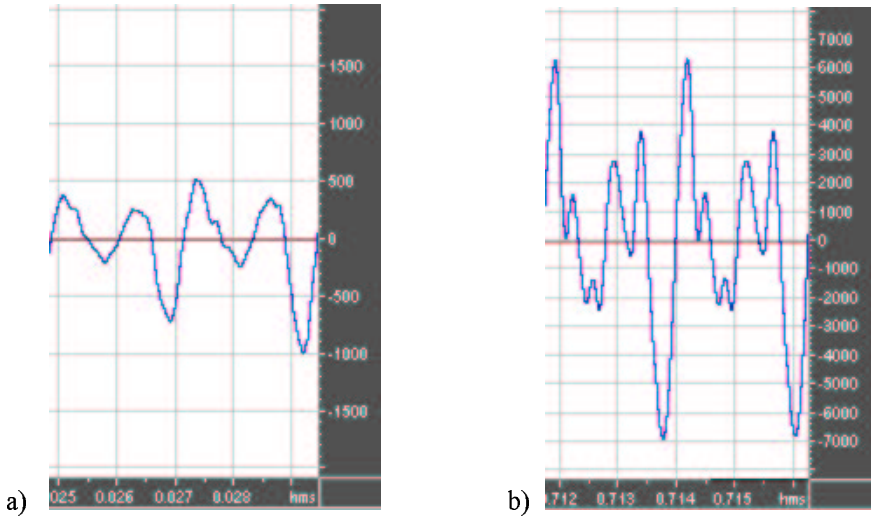

Figure 1. Time domain for a fragment corresponding to 2 periods of the oboe sound $a^1 = 440$Hz during the attack of the sound (a) and the quasi-steady state (b).

table $\mathbf{A} = (U, A \cup \{d\})$, where each element $u \in U$ corresponds to a sound sample, each element $a \in A$ is a numeric feature corresponding to one of the sound descriptors and the decision attribute $d \notin A$ labels particular object (sound) with integer code adequate to the instrument. Hence, we need a framework for preprocessing original data, in particular, for extracting features most relevant to the task of the sound recognition.

The sound data are taken from MUMS audio CD's that contain samples of broad range of musical instruments, including orchestral ones, piano, jazz instruments, organ, etc. (Opolko and Wapnick, 1987). These CD's are widely used in musical instrument sound research (Cosi et al., 1994; Martin and Kim, 1998; Wieczorkowska, 1999b; Fujinaga and McMillan, 2000; Kaminskyj, 2000; Eronen and Klapuri, 2000), so they can be considered as a standard. The database consists of 667 samples of recordings, divided into the following 18 classes: violin vibrato, violin pizzicato, viola vibrato, viola pizzicato, cello vibrato, cello pizzicato, double bass vibrato, double bass vibrato, double bass pizzicato, flute, oboe, b-flat clarinet, trumpet, trumpet muted, trombone, trombone muted, French horn, French horn muted, and tuba.

4.2. Envelope descriptors

Attributes $a \in A$ can be put into $\mathbf{A} = (U, A \cup \{d\})$ in various ways. They can be based on analysis of various descriptors, their changes in time,

their mutual dependencies, etc. Let us begin with the following example of a new, temporal attribute. Consider a given sound sample, referred to as an object $u \in U$. We can split it onto, say, 7 intervals of equal width. Average values of amplitudes within these intervals are referred to as $Amp.1, \ldots, 7$. Sequence $\overrightarrow{Amp}(u) = \langle Amp.1(u), \ldots, Amp.7(u) \rangle$ corresponds to a kind of envelope, approximating the behavior of amplitude of each particular u in time. We can consider, e.g., Euclidean distance over the space of such approximations. Then we can apply one of basic clustering or grouping methods to find the most representative envelopes. In Figure 2 we show representative envelopes as centroids obtained from the algorithm dividing data onto 6 clusters.

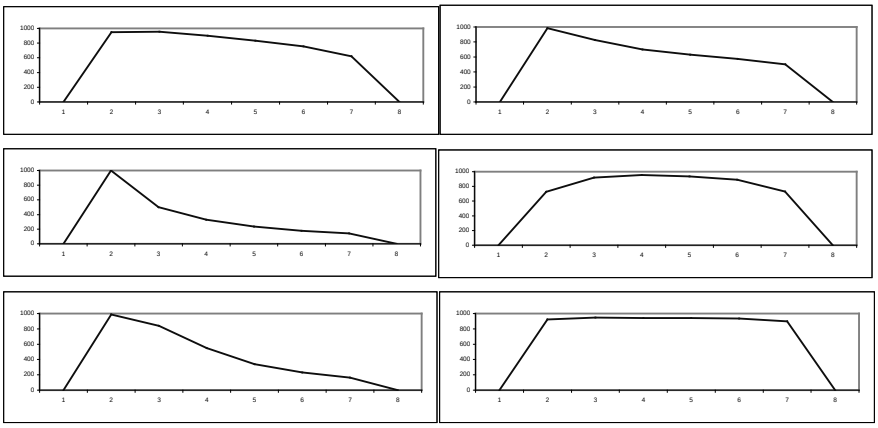

Figure 2. Centroids (the most typical shapes) of sound envelopes, used in clustering.

We obtain a new group of attributes, labeling each sample-object $u \in U$ with the following amplitude envelope parameters.

Envelope descriptors:

- $Amp.1, \ldots, 7$: Average values of amplitudes within the considered 7 intervals, respectively
- $EnvFill$: Area under the curve of envelope, approximated by means of values $Amp.1, \ldots, 7$
- $Cluster$: Number of the closest of 6 representative envelope curves, shown in Figure 2.

4.3. The usage of the time windows

The main difficulty of sound analysis is that many useful attributes of sound are related to a fragment of the sound sample. E.g. spectrum-based attributes (tristimulus parameters, pitch, etc.) describe rather a selected time frame on which the spectrum was calculated than the whole sound. Moreover, these attributes may change from one segment of time to another. One can take a frame (windowed) from quasi-steady part of a sample and treat it as a representative of the whole sound, but in this case we may loose too much information about the sample.

We take into account both the sample based and the frame based attributes. We consider time frames of the length equal to four times the fundamental sound period. Since the frame length is adjusted to the period, no special windowing is needed and we can simply use rectangular window. Within each window, we can calculate local values of spectral (and also other) descriptors. For each attribute, its local frame based values create the time series, which can be further analyzed in various ways. For instance, we can consider envelope based attributes similar to those introduced in the previous subsection. Such envelopes, however, would be referring not to the amplitudes, but to the dynamics of changes observed for spectral descriptors in time.

Usually, the frame length is constant for the whole analysis, with the most common length about 20-30 ms. For instance, Eronen and Klapuri (2000) used 20 ms frame, whereas Brown (1999) reported 23 ms analyzing frame. Batlle and Cano (2000) applied 25 ms frame, and Brown et al. (2001) used 32 ms frame. Such a frame is sufficient for most sounds, since it contains at least a few periods of the recorded sounds. However, it is too short for analysis of the lowest sounds we used, and too long for analysis of short pizzicato sounds, where changes are very fast, especially in case of higher sounds. This is why we decided to set up the length of those frames as four times the fundamental period of the sound. We decompose each musical sound sample into such frames and calculate value sequences and final features for each of them.

4.4. Fundamental frequency approximation

Spectral time series should be stored within an additional table, where each record corresponds to a small window taken from a sound sample. Hence, we first need to extract the lengths of windows for particular sounds. It corresponds to the well known problem of extracting fundamental frequency from data. Given frequency, we could calculate, for each particular sound sample, fundamental periods and derive necessary window based attributes.

There are numerous mathematical approaches to approximation of fundamental signal frequency by means of the frequency domain or estimation of the length of period (and fundamental frequency as an inverse) by means of the time domain. The methods used in musical frequency tracking include autocorrelation, maximum likelihood, cepstral analysis, Average Magnitude Difference Function (AMDF), methods based on zero-crossing of the sound wave etc., see (Brown and Zhang, 1991; Doval and Rodet, 1991; Cook et al., 1992; Beauchamp et al., 1993; Cooper and Ng, 1994; de la Cuadra et al., 2001). Most of the frequency tracking methods originate from speech processing. Methods applied to musical instrument sounds are usually tuned to the characteristics of spectrum (sometimes some assumptions about the frequency are required), and octave errors are common problem here. Therefore, the instrument-independent frequency estimation is quite difficult.

For our research purposes, we have used AMDF in the following form (Cook et al., 1992):

$$AMDF(i) = \frac{1}{N} \sum_{k=0}^{N} |A_k - A_{i+k}| \tag{4}$$

where N is the length of interval taken for estimation and A_k is the amplitude of the signal. The period for a given sound is determined as i corresponding to the minimal value of $AMDF(i)$. One can calculate the values of (4) within the interval of the first few admissible period lengths, and then approximate the period. The problem is that during the attack time the values of $AMDF$ are less reasonable than in case of the rest of the signal, after stabilization. We cope with this problem by evaluating the approximate period length within the stable part of the sound and then – tuning it with respect to the part corresponding to the attack phase (Wieczorkowska, 1999a). In our experiments we used a mixed approach to approximate periods, based both on searching for stable minima of $AMDF$ and maxima of spectrum obtained using DFT (Discrete Fourier Transform).

4.5. The final structure of the database

Given a method for extracting frequencies, we can accomplish the preprocessing stage and concentrate on data reduction and transformation. The obtained database framework is illustrated in Figure 3.

- Table SAMPLE (667 records, 14 columns) gathers values of temporal and spectral descriptors. Column *Instrument* states the code of musical instrument, together with its articulation (18 values).

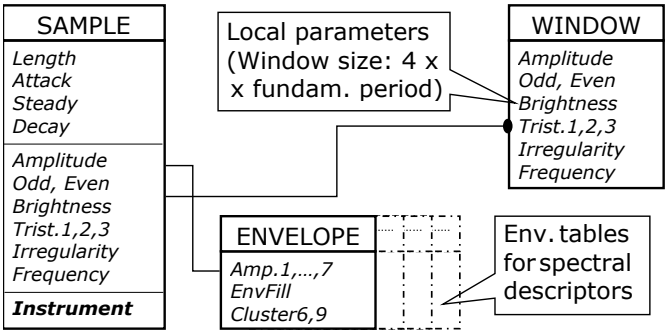

Figure 3. Relational musical sound database after the preprocessing stage.

- Table ENVELOPE (667 records, 10 columns) is linked in 1:1 manner with table SAMPLE. Its columns correspond to attributes derived by analyzing amplitudes, as in Subsection 4.2. However, we can define analogous ENVELOPE tables also for other descriptors.

- Table WINDOW (190800 records, 10 columns) gathers records corresponding to local time windows. According to the previous subsection, for each sample we obtain (Length·Frequency/4) records. Each record is labeled with spectral descriptors defined in the same way as for table SAMPLE, but calculated locally.

As a result, we obtain the relational database, where tables SAMPLE and WINDOW are linked in 1:n manner, by the code of the instrument sample (primary key for SAMPLE and foreign key for WINDOW). All additional data tables used in our experiments were derived from this main relational database. E.g. we have created (and collected in additional data table) envelopes of some spectral features of sound (see Section 7) by calculating average values of subsequent records from WINDOW table in 7 intervals of equal width.

5. Automatic extraction of new attributes

5.1. Relational approach

Given the database structure as illustrated by Figure 3, one can move to the next stage of the knowledge discovery process – data transformation. In the particular case of this application, we need features – attributes – describing the sound samples. Hence, we need to use information stored within the database to create new columns within

the table SAMPLE, where each record corresponds to the whole sound sample.

One can see that the ENVELOPE tables considered in the previous subsection consist of attributes describing musical samples. The values of these attributes were extracted from the raw data at the preprocessing level. Still, some new attributes can be added in a more automatic way. Namely, one can use relations between already existing tables. One can construct a parameterized space of possible features based on available relations. Then, one can search for optimal features in an adaptive way, verifying which of them seem to be the best for constructing decision models.

Such a process has been already implemented for SQL-like aggregations in (Wróblewski, 2000; Wróblewski, 2001b). Exemplary features, found automatically as SQL-like aggregations from table WINDOW, can be of the following nature:

- *average of LocIrr from WINDOW*
- *sum of LocTri3 from WINDOW where LocTri3 < LocTri2*

E.g., *average of LocIrr* is the new attribute with values equal to the average values of *irregularity* parameters (Section 2.3) calculated over the whole set of time windows corresponding to particular samples.

The goal of the searching algorithm is to extract aggregations of potential importance while distinguishing instrument decision classes. The example presented above collects the best new attributes, according to quality measures presented in Subsection 5.3. Such attributes are then added as new columns to table SAMPLE. The advantage of the presented approach is visible when using rule based data mining methods (rough set based data mining algorithm, see (Ślęzak, 2001; Wróblewski, 2001b)). Still, adding 9 new attributes to the original data table increased the recognition rate only to 49.7%, comparing with 48.5% obtained with the same algorithm originally. The problem is that, in this particular case of database with only one 1:n relation, the usage of the above approach does not provide much valuable knowledge. Moreover, the temporal nature of table WINDOW requires a slightly different approach than that directly based on SQL-like aggregations. We go back to this issue in Section 6.

5.2. Linear combinations

Automatic extraction of significant new features is possible also for single data tables, not embedded into any relational structure. In case of numerical features, such techniques as discretization, hyperplanes,

clustering and principal component analysis (Nguyen, 1997; Nguyen, 2000), are used to transform the original domains into more general or more descriptive ones. One can treat the analysis process over transformed data either as a modeling of a new data table (extended by new attributes given as a function of original ones) or, equivalently, as an extension of model language. The latter means, e.g., change of metric definition in k-NN algorithm or extension of language of rules.

In our approach the original data set is extended by a number of new attributes defined as linear combinations of the existing ones. Let $B = \{b_1, \ldots, b_m\} \subseteq A$ be a subset of attributes, $|B| = m$, and let $\alpha = (\alpha_1, \ldots, \alpha_m) \in \mathbf{R}^m$ be a vector of coefficients. Let $h : U \to \mathbf{R}$ be a function defined as:

$$h(u) = \alpha_1 b_1(u) + \ldots + \alpha_m b_m(u) \tag{5}$$

Usefulness of new attribute h depends on proper selection of parameters B and α. It is useful, when the data model (e.g. defined by decision rules) based on discretized values of h becomes more general, without loss of accuracy.

5.3. Quality measures

The evolution strategy algorithm can optimize the coefficients of h using various quality functions. Three of them are implemented in the current version of the Rough Set Expert System RSES (Bazan et al., 2002). Theoretical foundations of their usage are described in (Ślęzak and Wróblewski, 1999), as well as (Ślęzak, 2001; Wróblewski, 2001b).

Let L be the straight line in $\mathbf{R}^m$ defined by a given linear combination h. The general idea of the mentioned measures is given below.

Distance measure is an average (normalized) distance of objects from different decision classes in terms of h (i.e. projected onto L). Its value can be expressed as follows:

$$Dist(h) = \sum_{i=1,\ldots,N} \sum_{j=i+1,\ldots,N : d(u_i) \neq d(u_j)} \frac{|h(u_i) - h(u_j)|}{max(h) - min(h)} \tag{6}$$

where $max(h)$ and $min(h)$ are maximal and minimal values of h over objects $u \in U$. In (Ślęzak, 2001) it is shown that $Dist(h)$ is equal to the average rough set based quality of cuts defined on h.

Discernibility measure takes into account two components: distance (as above) and average discernibility of objects with different decision values. *Dist* turned out to be effective for classification of the benchmark data sets in (Ślęzak and Wróblewski, 1999). In its simplified form,

without considering the distance coefficient, it can be defined as follows (Ślęzak, 2001):

$$Disc(h) = \sum_{i=1}^{M} r_i^2 \tag{7}$$

where $r_i = |\{u \in U : c_i < h(u) \leq c_{i+1}\}|$ is the number of objects included in the i-th interval, for the minimal possible set of cuts $C_h = \{c_1, \ldots, c_M\}$, which enables to construct deterministic decision rules based on h. $Disc$ refers to intuition that a model with lower number of decision rules should be regarded as better than the others (Bazan et al., 2000; Pawlak, 1991).

Predictive measure is an estimate of the expected classifier's prediction quality when using only h. It is constructed with use of the probabilistic methods for approximating the expected values of coverage and sensibility (Wróblewski, 2001a; Wróblewski, 2001b). It is defined as

$$Pred(h) = 1 - \prod_{i=1}^{M} \left(1 - \frac{r_i - 1}{|U| - 1}\right) \tag{8}$$

$Pred$ is more suitable for the rule based data mining methods rather than for distance based ones (e.g. k-NN).

6. Temporal descriptors

6.1. Temporal vs. relational features

Extraction of temporal patterns or temporal clusters can be regarded as a special case of using 1:n connection between data tables. Here, new aggregated columns are understood in terms of deriving descriptors corresponding to trends in behavior of values of some locally defined columns (belonging to table WINDOW). Temporal aggregations cannot be expressed in SQL-like language. One of the main future goals is to automatize the process of defining temporal attributes, to get ability of massive search through the space of all possibilities of temporal descriptors. Then, one would obtain an extended model of relational feature extraction developed in (Wróblewski, 2000) and (Wróblewski, 2001b), meeting the needs of the modern database analysis.

We propose to search for temporal patterns that can potentially be specific for one instrument or a group of instruments. Such patterns can be further used as new descriptors like *Cluster* in table ENVELOPE. These attributes describe general trends of the amplitude values in time. Results in Section 7 show potential importance of such features.

Similar analysis can be performed over spectral features stored in table WINDOW, by searching for, e.g., *temporal templates* (Synak, 2000).

6.2. Temporal patterns

Generation of temporal patterns requires the choice of descriptors that would be used to characterize sound samples and a method to measure values of those descriptors in time. We use the time window based technique. We browse a sample with time windows of certain size. Within a given window we compute all descriptor values and generate one object of a *temporal information system* $\mathbf{A} = (\{x_1, x_2, \ldots, x_n\}, A)$, where x_i is a measurement from the i-th window using descriptors from A (this way we constructed table WINDOW). Next, we use it to determine optimal *temporal templates* that correspond to temporal patterns. Thus for one sample we compute a sequence of temporal templates.

Temporal templates are built from descriptors $(a \in V)$, where $a \in A$, $V \subseteq V_a$ and $V \neq \emptyset$. Formally, *template* is a set of descriptors involving any subset $B \subseteq A$:

$$T = \{(a \in V) : a \in B, V \subseteq V_a\}. \tag{9}$$

By *temporal template* we understand

$$\mathbf{T} = (T, t_s, t_e), \quad 1 \leq t_s \leq t_e \leq n, \tag{10}$$

that is template placed in time – with corresponding period $[t_s, t_e]$ of occurrence (see Figure 4).

Let us define some basic notions related to temporal templates. First of all, by *width* we understand the length of period of occurrence, i.e. $width(\mathbf{T}) = t_e - t_s + 1$. *Support* is the number of objects from period $[t_s, t_e]$ matching all descriptors from T. Finally, *precision* of temporal template is defined as a sum of precisions of all descriptors from T, where precision of descriptor $(a \in V)$ is given by:

$$Precision\,((a \in V)) = \begin{cases} \frac{card(V_a) - card(V)}{card(V_a) - 1} & card(V_a) \geq 1 \\ 1 & otherwise \end{cases} \tag{11}$$

We consider quality of temporal template as a function of width, support and precision. Templates and temporal templates are intensively studied in literature, see (Agrawal et al., 1996; Nguyen, 2000; Synak, 2000). To outline the intuition, behind these notions, let us understand template as a strong regularity in data, whereas temporal template as strong regularity occurring in time.

	A	a	b	c	d	e
	x_1	·	·	·	·	·
$\mathbf{T}_1$	x_2	*u*	·	*v*	·	·
	x_3	*u*	·	*v*	·	·
	x_4	·	·	·	·	·
	x_5	*u*	·	*v*	·	·
	x_6	*u*	·	*v*	·	·
	x_7	·	·	·	·	·
	x_8	*u*	·	*v*	·	·
	x_9	·	·	·	·	·
$\mathbf{T}_2$	x_{10}	·	*x*	·	*y*	·
	x_{11}	·	*x*	·	*y*	·
	x_{12}	·	·	·	*y*	·
	x_{13}	·	*x*	·	*y*	·
	x_{14}	·	*x*	·	·	·
	x_{15}	·	·	·	·	·

Figure 4. Temporal templates for the system $\mathbf{A} = (\{x_1, \ldots, x_{15}\}, \{a, b, c, d, e\})$: $\mathbf{T_1} = (\{(a \in \{u\}), (c \in \{v\})\}, 2, 8)$, $\mathbf{T_2} = (\{(b \in \{x\}), (d \in \{y\})\}, 10, 13)$.

6.3. Episodes

In one musical sound sample we can find several temporal templates. They can be time dependent, i.e. one can occur before or after another. We can treat them as a sequence of events, consistent with the characteristics of the evolution of the sound descriptors in time. Depending on the needs, we can represent such sequences purely in terms of the temporal template occurrence in time, as illustrated in Figure 5, or by focusing on the entire specifications of templates, as in Figure 6.

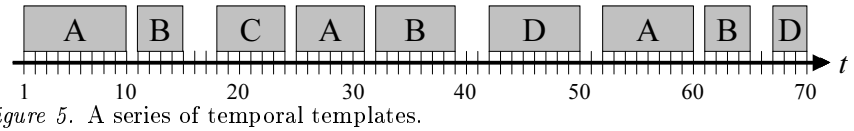

Figure 5. A series of temporal templates.

From sequences of templates we can discover frequent *episodes* – collections of templates occurring together (Mannila et al., 1998; Synak, 2000). An episode occurs in a sequence of templates if each element (template) of episode exists in a sequence and order of occurrence is preserved. For example, episode AAC occurs in sequence $B\underline{A}C\underline{A}BB\underline{C}$. We expect some of such episodes to be specific only for particular instrument or group of instruments.

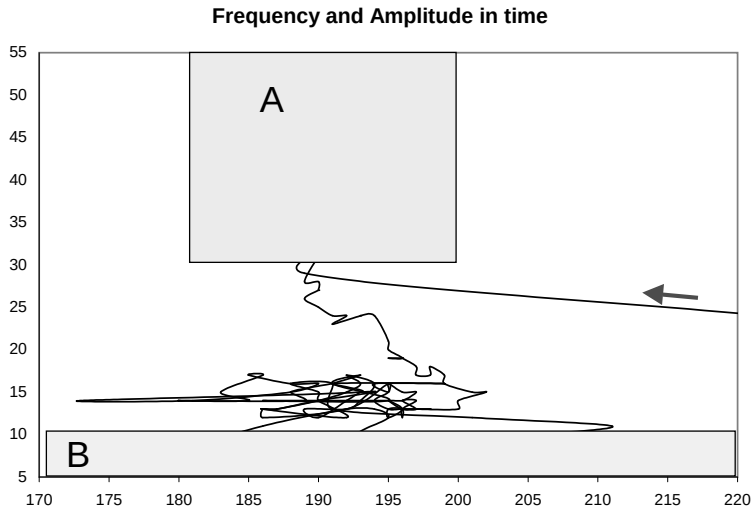

Figure 6. Evolution of *Frequency* and *Maximal Amplitude* in time. Examples of templates: A= $\{(Frequency \in [180, 200]), (Maximal\,Amplitude \in [30, 60])\}$, B= $\{(Maximal\,Amplitude < 10)\}$. The episode ABB occurs in this sound.

6.4. The extraction process

We propose to construct the new sound descriptors based on episodes. Below we present the scheme of their generation:

STEP 1: For each training sample we generate a sequence of temporal templates. As the input we take objects from table WINDOW, i.e. sound descriptors of samples measured in windows of size equal to four times the fundamental period of sound. Because these attributes are real-valued, we can discretize them first, by using the uniform scale quantization (the number of intervals is a parameter).

STEP 2: Number of different templates (in terms of descriptors only) found for all samples is relatively large. Therefore, we propose to search for the template dependencies, characteristic for samples of particular decision classes, by basing only on representative templates. Such templates can be generated as optimal with respect to some quality measures. The following measures are just examples based on theories of machine and statistical learning, as well as rough sets, which can be applied at this stage of the process. Measure

$$BayesDist(T) = \sum_{v_d \in V_d} |P(T/v_d) - P(T)| \tag{12}$$

describes the impact of decision classes onto probability of T. By $P(T)$ we mean simply the prior probability that elements of the universe satisfy T, estimated directly from data. By $P(T/v_d)$ we mean the same probability, but derived from the decision class corresponding to $v_d \in V_d$. If we regard T as the left side of an inexact decision rule $T \Rightarrow d = v_d$, then $P(T/v_d)$ describes its sensitivity (Mitchell, 1998). Quantities $|P(T/v_d) - P(T)|$ express a kind of degree of information we gain about T given knowledge about membership of the analyzed objects to particular decision classes. According to the Bayesian reasoning principles (Box and Tiao, 1992), $BayesDist(T)$ provides the degree of information we gain about decision probabilistic distribution, given additional knowledge about satisfaction of T. The second exemplary measure

$$RoughDisc(T) = P(T)(1 - P(T)) - \sum_{v_d \in V_d} P(T, v_d)(P(v_d) - P(T, v_d)) \tag{13}$$

is an adaptation of one of the rough set measures used e.g. in (Nguyen, 1997) and (Ślęzak, 2001) to express the number of pairs of objects belonging to different decision classes, being discerned by a specified condition. Normalization, understood as division by the number of all possible pairs, results with (13), where $P(T)$ is understood as before, $P(v_d)$ denotes the normalized cardinality of the decision class corresponding to $v_d \in V_d$ and $P(T, v_d)$ is the joint probability of satisfying conditions T and $d = v_d$.

STEP 3: Using template representatives we replace each found template with the closest representative. The exemplary measure of closeness can be defined as follows:

$$DIST(T_1, T_2) = \sum_{a \in A} \left(1 - \frac{|V_a^{T_1} \cap V_a^{T_2}|}{|V_a^{T_1} \cup V_a^{T_2}|}\right). \tag{14}$$

where $V_a^{T_i}$, $i = 1, 2$, is the subset of values of a, corresponding to generalized descriptor ($a \in V_a^{T_i}$), which is a component of T_i with respect to the attribute a.

STEP 4: Each sequence of temporal templates, found for each sample, is now expressed in terms of a collection of representative templates. We can expect some regularities in those sequences, possibly specific for one or more classes of instruments. To find those regularities we propose an algorithm, based on A-priori method (Agrawal and Srikant, 1994; Mannila et al., 1998), which discovers frequently occurring episodes with respect to some occurrence measure Occ. The difference, comparing e.g. to Winepi algorithm (Mannila et al., 1998), is that here

we are looking for episodes across many series of events. On the input of the algorithm we have a set of template sequences and the threshold τ.

Frequently occurring episode detection

1. $\mathcal{F}_1 = \{$frequently occurring 1-sequences$\}$

2. for $(l = 2;\ \mathcal{F}_{l-1} \neq \emptyset;\ l{+}{+})$ {

3. $\quad \mathcal{C}_l = GenCandidates(\mathcal{F}_1, \mathcal{F}_{l-1}, l)$

4. $\quad \mathcal{F}_l = \{c \in \mathcal{C}_l : Occ(c) \geq \tau\}$ }

5. return $\bigcup_l \mathcal{F}_l$

At first, we check which templates occur frequently enough in all sequences. That forms the set $\mathcal{F}_1$ of frequent episodes of length one. We can consider several measures of occurrence. The fundamental one is just the number of occurrences, however, by being "frequent" we can also understand frequent occurrence in one class of instruments and rare occurrence in another classes. Therefore, we can adapt measures (12), (13) to the definition of function $Occ()$. Next, we recursively create a set of candidates $\mathcal{C}_l$ by combining frequent templates ($\mathcal{F}_l$) with frequently occurring episodes of size $l-1$ ($\mathcal{F}_{l-1}$). The last step is to verify the set of candidates $\mathcal{C}_l$ and eliminate infrequent episodes.

STEP 5: We generate just two attributes related to the occurrence of frequent episodes in a series of templates found in a sound sample. The first one is an episode of highest value of function $Occ()$ out of all episodes that occur in a given sequence. The second one is the longest episode – if there are more than one, we choose the episode with higher value of $Occ()$.

The presented method requires evaluation of many parameters. The most important ones are: window size (when generating temporal templates), number of representative templates, quality and frequency measure, and frequency threshold. After all, we concentrate just on two episode based attributes defined in the above Step 5. In Subsection 7.2, one can see that they can improve the classification results a lot. Further research is needed to establish a broader family of valuable episode based descriptors.

7. Results of experiments

7.1. Published results

The research on musical instrument sound classification is performed all over the world. However, the data sets differ from one experiment to another and it is almost impossible to compare the results. The most common data come from McGill University Master Samples (MUMS) CD collection (Opolko and Wapnick, 1987), i.e. the recordings that we also used in our research.

Experiments carried out so far operate on various number of instruments and classes. Some experiments are based on a few instruments, and sometimes only singular sounds of the selected instruments are used. It is also quite common to classify not only instrument, or instrument and a specific articulation, but also instrument categories. The final results vary depending on the size of the data, feature vector, as well as classification and testing methods applied.

Brown et al. (2001) reported correct identifications of 79%–84% for 4 classes (oboe, sax, clarinet and flute), with cepstral coefficients, constant-Q coefficients, and autocorrelation coefficients applied to short segments of solo passages from real records. Each instrument was represented by at least 25 sounds. The results depended on the chosen training sounds and the number of clusters used. Bayes decision rules were applied to the data clustered using k-means algorithm. This method was earlier applied by Brown (1999) to oboe and sax data only.

Kostek and Czyzewski (2001) applied 2-layer feedforward neural networks with momentum method to classify various groups of 4 orchestral instruments, recorded on DAT. Feature vectors consisted of 14 FFT-based or 23 wavelet-based parameters. The results reached 99% for FFT vectors and 91% for wavelet vectors (Various testing procedures were applied).

Martin and Kim (1998) identified instrument families (string, woodwind and brass) with approximately 90% performance and individual instruments with an overall success rate of approximately 70%, for 1023 isolated tones over the full pitch ranges of 14 orchestral instruments. The classifiers were constructed based on Gaussian models, arrived at through Fisher multiple-discriminant analysis, and cross-validated with multiple 70%/30% splits. 31 perceptually salient acoustic features, related to the physical properties of source excitation and resonance structure were calculated for MUMS sounds.

Wieczorkowska (1999a), (1999b) applied rough set based algorithms, decision trees and some other algorithms to the data representing 18 classes (11 orchestral instruments, full pitch range), taken from MUMS

CDs. The results approached 90% for instrument families (string, woodwind, brass) and 80% for singular instruments with specified articulation. Various testing procedures were used, including 70%/30% and 90%/10% splits. Feature vectors were based on FFT and wavelet analysis, including time-domain features as well.

Fujinaga and MacMillan (2000) reported the recognition rate 50% for the 39-timbre group (23 orchestral instruments) with over 1300 notes from McGill CD library and 81% for a 3-instrument group (clarinet, trumpet, bowed violin). They applied k-NN classifier and a genetic algorithm to seek the optimal set of weights for the features. Standard leave-one-out procedure was used to calculate the recognition rate.

Kaminskyj (2000) obtained overall accuracy of 82% using combined k-NN classifiers with different k values and using the leave-one-out classification scheme. The data described 19 musical instruments of definite pitch, taken from MUMS CDs. Kaminskyj used the following features: RMS amplitude envelope, constant Q transform frequency spectrum and multidimensional scaling analysis trajectories.

Eronen and Klapuri (2000) recognized instrument families (string, brass, and woodwind) with 94% accuracy and individual instruments with 80% rate using 1498 samples covering full ranges of 30 orchestral instruments, played with various articulation techniques. 44 spectral and temporal features were calculated for sounds mostly taken from MUMS collection, and guitar and piano sounds by amateur players recorded on DAT. Gaussian and k-NN classifiers were used, and cross-validation with 70%/30% splits of train and test data was performed.

7.2. Experiments based on the proposed approaches

Extensive comparison of the results of experiments on musical instrument sound classification worldwide is presented in (Herrera et al., 2000). Our classification results are presented in Table I. We performed our experiments for 18 decision classes, using standard CV-5 method for evaluation of resulting decision models.
Our results correspond to two approaches to constructing classifiers:

- Best k-NN: Standard implementation with tuning parameter k. The best results among different values of k as well as different metrics (Euclidean, Manhattan) are presented.
- RS-decision rules: Algorithm presented in (Bazan et al., 2000) for finding optimal ensembles of decision rules, based on the theory of rough sets (Pawlak, 1991).

Particular rows of the table in Table I correspond to performance of the above algorithms over decision tables consisting of various sets

Table I. Experimental results (classification correctness).

Attributes	**Best k-NN**	**RS-decision rules**
Envelope	36.3%	17.6%
Envelope with linear combinations	42.1%	11.1%
Temporal	54.3%	39.4%
Spectral	34.2%	14.6%
Spectral + Temporal Patterns	–	58.9%
Temporal + Spectral	68.4%	48.5%
Temporal + Spectral + Relational	64.8%	49.7%
Spectral envelopes	32.1%	–
Clustered spectral envelopes	31.3%	–

of conditional attributes. Groups of features correspond to notation introduced in Section 4:

- Envelope: 36% of correct classification of new cases into 18 possible decision classes – a good result in case of k-NN over several quite naive conditional features.

- Envelope with linear combinations: Improvement of correct classification in case of k-NN after adding to previous envelope features some linear combinations, found by the approach discussed in Section 5. This confirms the thesis about importance of searching for optimal linear combinations over semantically consistent original features, stated in (Ślęzak and Wróblewski, 1999). On the other hand, one can see that extension of the set of envelope based attributes is not good in combination with RS-decision rules – 11.1% is not much better than random choice.

- Temporal: Very good result for just a few, very simple descriptors, ignoring almost the whole knowledge concerning the analysis of music instrument sounds. Still k-NN (54.3%) performs better than RS-decision rules (39.4%). In general, one can see that k-NN is a better approach for this specific data (although it's not always the case – see e.g. (Polkowski and Skowron, 1998)). Obviously, it would be still better to base, at least partially, on decision rules while searching for intuitive explanation of the reasoning process.

- Spectral: Classical descriptors related to spectrum analysis seem to be insufficient to this type of task. From this perspective, the results obtained for Temporal features are even more surprising.

- Spectral + Temporal Patterns: Using methodology of the temporal patterns extraction (see Section 6), we created two new attributes that were added to Spectral features. As because they are not numerical ones, we performed experiments by application of RS-decision rules only. We can observe improvement of classification results to 58.9%.

- Temporal + Spectral: Our best result, 68.4% for k-NN, still needs further improvement. Again, performance of RS-decision rules is worse (48.5%), although other rough set based methods provide better results – e.g., application of the algorithm for the RSES library (see (Bazan and Szczuka, 2000)) gives 50.3%.

- Temporal + Spectral + Relational: Another rough set based classification algorithms, described in (Ślęzak, 2001) and (Wróblewski, 2001b), yield – if taken together with new (automatically created) features listed in Section 5 – up to 49.7%.

- Spectral envelopes: A general shape (calculated over 6 intervals and normalized) of change of spectral parameters in time. There are 5 spectral features (*Brightness*, *Irregularity*, *Tristimulus1,2,3*) which evolution is described by 30 numerical values. Relatively low recognition rate (32.1%), especially compared with the result for amplitude envelope (only 6 numerical values), shows that changes of spectral features are not specific enough. Optimized linear combinations of these 30 numerical values give the same recognition quality, with a number of attributes limited to 15. Initial results of experiments using rule based system were not promising (below 15%, probably because all of these features are numerical) and this method was not used in the further experiments.

- Clustered spectral envelopes: The 5 envelopes used in the previous experiment were clustered (into 5 groups each), then a distance to a centroid of each cluster was calculated. These distances (5 numerical values for each spectral attribute) were collected in a decision table described by 25 conditional attributes. Results (and discussion) are similar to the previous experiment.

8. Conclusions

We focus on methodology of musical instrument sound recognition, related to KDD process of the training data analysis. Our classification is based on appropriately extracted features, calculated for particular sound samples – objects in a relational database of the sound sample representations. We use features similar to descriptors from MPEG-7, but we also consider the clustering and time series framework, by taking as new descriptors temporal patterns observed for particular features. This is a novel approach, a step towards automatic extraction of musical information within multimedia contents.

The most important for further research is to perform more experiments with classification of new cases, basing on decision models derived from training data in terms of the introduced data structure. It seems that the need of transformation is obvious in case of attributes which are neither numerical nor discrete, e.g. when objects are described by time series. In the future we plan to combine the clustering methods with time trends analysis, to achieve an efficient framework for expressing the dynamics of the changes of complex feature values in time.

Acknowledgements

This research was sponsored by the Research Center of PJIIT, supported by the Polish National Committee for Scientific Research (KBN).

References

Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., and Verkamo, I. (1996). Fast Discovery of Association Rules. In *Proc. of the Advances in Knowledge Discovery and Data Mining* (pp. 307–328). AAAI Press / The MIT Press, CA.

Agrawal, R., Srikant, R. (1994). Fast Algorithms for Mining Association Rules. In *Proc. of the VLDB Conference*, Santiago, Chile.

Ando, S. and Yamaguchi, K. (1993). Statistical Study of Spectral Parameters in Musical Instrument Tones. *J. Acoust. Soc. of America*, 94(1), 37–45.

Batlle, E. and Cano, P. (2000). Automatic Segmentation for Music Classification using Competitive Hidden Markov Models. *Proceedings of International Symposium on Music Information Retrieval.* Plymouth, MA. Available at http://www.iua.upf.es/mtg/publications/ismir2000-eloi.pdf.

Bazan, J. G., Nguyen, H. S., Nguyen, S. H, Synak, P., and Wróblewski, J. (2000). Rough Set Algorithms in Classification Problem. In L. Polkowski, S. Tsumoto, and T.Y. Lin (Eds.), *Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems.* Physica-Verlag, 49–88.

Bazan, J. G. and Szczuka, M. (2000). RSES and RSESlib - A collection of tools for rough set computations. In W. Ziarko and Y. Y. Yao (Eds.), *Proc. of RSCTC'00*, Banff, Canada. See also: http://alfa.mimuw.edu.pl/~rses/.

Bazan, J. G., Szczuka, M., and Wróblewski, J. (2002). A New Version of Rough Set Exploration System. In *Proc. of RSCTC'02*. See also: http://alfa.mimuw.edu.pl/~rses/.

Beauchamp, J. W., Maher, R., and Brown, R. (1993). Detection of Musical Pitch from Recorded Solo Performances. 94th AES Convention, preprint 3541, Berlin.

Box, G. E. P., and Tiao, G. C. (1992). Bayesian Inference in Statistical Analysis. Wiley.

Brown, J. C. (1999). Computer identification of musical instruments using pattern recognition with cepstral coefficients as features. *J. Acoust. Soc. of America*, 105, 1933–1941.

Brown, J. C. and Zhang, B. (1991). Musical Frequency Tracking using the Methods of Conventional and 'Narrowed' Autocorrelation. *J. Acoust. Soc. Am.*, 89, 2346–2354.

Brown, J. C., Houix, O., and McAdams, S. (2001). Feature dependence in the automatic identification of musical woodwind instruments. *J. Acoust. Soc. of America*, 109, 1064–1072.

Cook, P. R., Morrill, D., and Smith, J. O. (1992). An Automatic Pitch Detection and MIDI Control System for Brass Instruments. Invited for special session on Automatic Pitch Detection, Acoustical Society of America, New Orleans.

Cooper, D. and Ng, K. C. (1994). A monophonic pitch tracking algorithm. Available at http://citeseer.nj.nec.com/cooper94monophonic.html.

Cosi, P., De Poli, G., and Lauzzana, G. (1994). Auditory Modelling and Self-Organizing Neural Networks for Timbre Classification *Journal of New Music Research*, 23, 71–98.

de la Cuadra, P., Master, A., and Sapp, C. (2001). Efficient Pitch Detection Techniques for Interactive Music. *ICMC*. Available at http://www-ccrma.stanford.edu/ pdelac/PitchDetection/icmc01-pitch.pdf.

Doval, B. and Rodet, X. (1991). Estimation of Fundamental Frequency of Musical Sound Signals. *IEEE*, A2.11, 3657–3660.

Düntsch I., Gediga G., and Nguyen H. S. (2000). Rough set data analysis in the KDD process. In *Proc. of IPMU 2000*, 1, (pp. 220–226). Madrid, Spain.

Eronen, A. and Klapuri, A. (2000) Musical Instrument Recognition Using Cepstral Coefficients and Temporal Features. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2000* (753–756). Plymouth, MA.

Fujinaga, I. and McMillan, K. (2000). Realtime recognition of orchestral instruments. *Proceedings of the International Computer Music Conference* (141–143).

Herrera, P., Amatriain, X., Batlle, E., and Serra X. (2000). Towards instrument segmentation for music content description: a critical review of instrument classification techniques. In *Proc. of International Symposium on Music Information Retrieval (ISMIR 2000)*, Plymouth, MA.

ISO/IEC JTC1/SC29/WG11 (2002). MPEG-7 Overview. Available at http://mpeg.telecomitalialab.com/standards/mpeg-7/mpeg-7.htm.

Kaminskyj, I. (2000). Multi-feature Musical Instrument Classifier. *MikroPolyphonie* 6 (online journal at http://farben.latrobe.edu.au/).

Kostek, B. and Czyzewski, A. (2001). Representing Musical Instrument Sounds for Their Automatic Classification. *J. Audio Eng. Soc.*, 49(9), 768–785.

Kostek, B. and Wieczorkowska, A. (1997). Parametric Representation of Musical Sounds. *Archive of Acoustics*, 22(1), Institute of Fundamental Technological Research, Warsaw, Poland, 3–26.

Lindsay, A. T. and Herre, J. (2001). MPEG-7 and MPEG-7 Audio – An Overview. *J. Audio Eng. Soc.*, 49(7/8), 589–594.

Liu, H. and Motoda, H. (Eds.) (1998). *Feature extraction, construction and selection – a data mining perspective*. Kluwer Academic Publishers, Dordrecht.

Mannila, H., Toivonen, H., and Verkamo, A. I. (1998). Discovery of frequent episodes in event sequences. Report C-1997-15, University of Helsinki, Finland.

Martin, K. D. and Kim, Y. E. (1998). 2pMU9. Musical instrument identification: A pattern-recognition approach. 136-th meeting of the Acoustical Soc. of America, Norfolk, VA.

Mitchell, T. (1998). Machine Learning. Mc Graw Hill.

Nguyen, H.S.: Discretization od Real Value Attributes: Boolean Reasoning Approach. Rozprawa doktorska. Uniwersytet Warszawski (1997).

Nguyen H. S. (1997). Discretization od Real Value Attributes: Boolean Reasoning Approach. Ph.D. Dissertation, Warsaw University, Poland.

Nguyen S. H. (2000). Regularity Analysis And Its Applications In Data Mining. Ph.D. Dissertation, Warsaw University, Poland.

Opolko, F. and Wapnick, J. (1987). MUMS – McGill University Master Samples. CD's.

Pawlak, Z. (1991). *Rough sets – Theoretical aspects of reasoning about data*. Kluwer Academic Publishers, Dordrecht.

Peeters, G., McAdams, S., and Herrera, P. (2000). Instrument Sound Description in the Context of MPEG-7. In *Proc. International Computer Music Conf. (ICMC'2000)*, Berlin. Av. at http://www.iua.upf.es/mtg/publications/icmc00-perfe.pdf

Polkowski, L. and Skowron, A. (Eds.) (1998). *Rough Sets in Knowledge Discovery* **1**, **2**. Physica-Verlag, Heidelberg.

Pollard, H. F. and Jansson, E. V. (1982). A Tristimulus Method for the Specification of Musical Timbre. *Acustica*, 51, 162–171.

Ślęzak, D. (2001). Approximate decision reducts. Ph.D. thesis, Institute of Mathematics, Warsaw University.

Ślęzak, D., Synak, P., Wieczorkowska, A. A., and Wróblewski, J. (2002). KDD-based approach to musical instrument sound recognition. In M.-S. Hacid, Z. W. Ras, D. Zighed, and Y. Kodratoff (Eds.), *Foundations of Intelligent Systems* (pp. 29–37), LNCS/LNAI 2366, Springer.

Ślęzak, D. and Wróblewski, J. (1999). Classification algorithms based on linear combinations of features. In *Proc. of PKDD'99* (pp. 548–553). Praga, Czech Republik: LNAI 1704, Springer, Heidelberg. Available at http://www.mimuw.edu.pl/~jakubw/bib/.

Synak, P. (2000). Temporal templates and analysis of time related data. In W. Ziarko and Y. Y. Yao (Eds.), *Proc. of RSCTC'00*, Banff, Canada.

Toiviainen, P. (1996). Optimizing Self-Organizing Timbre Maps: Two Approaches. *Joint International Conference, II Int. Conf. on Cognitive Musicology* (pp. 264–271), College of Europe at Brugge, Belgium.

Wieczorkowska, A. A. (1999a). The recognition efficiency of musical instrument sounds depending on parameterization and type of a classifier. PhD thesis (in Polish), Technical University of Gdansk, Poland.

Wieczorkowska, A. (1999b). Rough Sets as a Tool for Audio Signal Classification. In Z. W. Ras, A. Skowron (Eds.), *Foundations of Intelligent Systems* (pp. 367–375). LNCS/LNAI 1609, Springer.

Wieczorkowska, A. A. and Raś, Z. W. (2001). Audio Content Description in Sound Databases. In N. Zhong, Y. Yao, J. Liu, and S. Ohsuga (Eds.), *Web Intelligence: Research and Development* (pp. 175–183). LNCS/LNAI 2198, Springer.

Wróblewski, J. (2000). Analyzing relational databases using rough set based methods. In *Proc. of IPMU'00* **1** (pp. 256–262), Madrid, Spain. Available at http://www.mimuw.edu.pl/~jakubw/bib/.

Wróblewski, J. (2001a). Ensembles of classifiers based on approximate reducts. Fundamenta Informaticae **47** (3,4), IOS Press (pp. 351–360). Available at http://www.mimuw.edu.pl/~jakubw/bib/.

Wróblewski, J. (2001b). Adaptive methods of object classification. Ph.D. thesis, Institute of Mathematics, Warsaw University. Available at http://www.mimuw.edu.pl/~jakubw/bib/.