Studies in Computational Intelligence, Volume 3

Editor-in-chief
Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
E-mail: kacprzyk@ibspan.waw.pl

Further volumes of this series
can be found on our homepage:
springeronline.com

Vol. 1. Tetsuya Hoya
Artificial Mind System – Kernel Memory Approach, 2005
ISBN 3-540-26072-2

Vol. 2. Saman K. Halgamuge, Lipo Wang (Eds.)
Computational Intelligence for Modelling and Prediction, 2005
ISBN 3-540-26071-4

Vol. 3. Bożena Kostek
Perception-Based Data Processing in Acoustics, 2005
ISBN 3-540-25729-2

Bożena Kostek

Perception-Based Data Processing in Acoustics

Applications to Music Information Retrieval and Psychophysiology of Hearing

Springer

Professor Bożena Kostek

Multimedia Systems Department
Faculty of Electronics,
Telecommunications and Informatics
Gdansk University of Technology
ul. Narutowicza 11/12
80-952 Gdansk, Poland
and
Institute of Physiology and Pathology of Hearing
ul. Pstrowskiego 1
01-943 Warsaw, Poland
E-mail: bozenka@sound.eti.pg.gda.pl
http://sound.eti.pg.gda.pl

Library of Congress Control Number: 2005926348

ISSN print edition: 1860-949X
ISSN electronic edition: 1860-9503
ISBN-10 3-540-25729-2 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-25729-5 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media
springeronline.com

Printed in The Netherlands

Typesetting: by the authors and TechBooks using a Springer LaTeX macro package

Printed on acid-free paper SPIN: 11412595 89/TechBooks 5 4 3 2 1 0

To my Father

FOREWORD

Recent years brought up many new techniques combining various aspects in so-called cognitive processing. This way of computing can be used with success in many areas of science and engineering by offering better analogy to human-like processing of information. Such an approach may be especially interesting in acoustics, where we deal with very inaccurate perceptions of phenomena due to the hearing sense characteristics which are highly imprecise with regard to time and spectral resolution.

This book demonstrates in which way soft computing methods (e.g. rough set-based methods) can be applied to provide flexible information processing capabilities for handling ambiguous decision making problems such as for examples musical signal processing, pattern classification, feature and rule generation. Methods of integrating rough sets, fuzzy and artificial neural networks for efficient knowledge discovery are also shown.

Fuzzy logic provides yet another tool that seems one of the best solutions for processing such inaccurate information as can be found in acoustics. In many domains building up membership functions could be problematic, however in acoustics the so-called subjective testing provides a good solution to this problem. Even, if such a testing is time consuming, it falls into the realm of human expertise, thus providing a class of perceptual membership functions. In addition, rules are quite obvious in the acoustic domain, and if not, they can be mined using other soft computing techniques, such as, for example, rough sets. In such a way it is possible to mimic human hearing perception and the way of processing perceived information in human brains.

The book addresses a number of topics such as the fundamentals of hearing and music perception, musical data representation and analysis, automatic classification of musical instrument sounds and musical phrases, musical sound separation, automatic recognition of musical styles, sound processing in hearing prostheses based on artificial neural networks, rough sets fuzzy logic principles, and others – based on cognitive approach. A review of soft computing and data mining techniques is provided, including all mentioned methods and others such as decision trees, evolutionary processing, and genetic algorithms. This book provides however a bal-

anced mixture of both theory and review of applications along with extensive bibliography.

The author has attempted to address a broad group of individuals with the common interest – music and hearing perception and sound processing based on cognitive approach. This is an important issue because of increasing specialization in science, however this book may be read by specialists from various domains and give them a comprehensive view on presented subjects.

This is a book of many new ideas and results, that will be of interest to all those interested in modern approach to imperfect data analysis. The author deserves highest appreciation for her valuable work.

Zdzisław Pawlak

PREFACE

The past can be pondered, the future must be created

E. Schillebeeckx

The emerging concept of human-centered computing or anthropomorphic approach represents a significant move towards intelligent systems, and affords a new perspective on information technology. Relationships between the human brain, mind and perception that have the potential of enhancing peoples' cognitive performance can be found in many domains, examples of which will be shown in relation to music processing and classification. On the other hand, it would be advisable to design systems capable of imitating perceptual processes that are best adapted to specific technological problems.

The objective of this monograph is to provide novel insights into perceptual mechanisms underlying the processing of sound and music in different environments. A solid understanding of these mechanisms is vital for numerous technological applications such as, for example, information retrieval from distributed musical databases. In order to investigate the cognitive mechanisms underlying music perception some soft computing methods will be used. The system proposed by the Author, based on the rough set method and fuzzy logic, provides knowledge on how humans internally represent such notions as quality and timbre and therefore it allows the human-like automatic processing of musical data. In addition, the automatically extracted knowledge on the above processes can be compared to fundamentals of hearing psychophysiology and to principles of music perception. Also other applications of hybrid decision systems to problem solving in music and acoustics will be exemplified and discussed in this book based not only on the review of some literature sources, but also on the experimental results obtained in the Multimedia System Department, Gdansk University of Technology.

The aim of this book is to show examples of the implementation of computational intelligence methods in musical signal and music analysis, as well as in the classification tasks. A part of this book contains a short review of perceptual bases of hearing and music. Then methods and techniques that can be classified as computational intelligence or machine-learning are shortly introduced. The presented methods are applied in the areas considered to be most relevant to music information retrieval (MIR) and acoustics. Accordingly, methods based on such learning algorithms as neural networks, rough sets, fuzzy-logic, and genetic algorithms were conceived, implemented and tested on musical data. In addition, the above-mentioned methods were applied to the analysis of musical duets, musical phrases and audio signals. Another problem discussed within the framework of this book is the 'computing with words' concept applied to both acoustics and psychophysiology. Perception-based analysis applied to psychophysiology focuses on the evaluation of hearing impairments. Application of neural networks to the processing of beamformer signals is another issue reviewed in this book. The last application described is devoted to the problem of audio-visual correlation search. This is based on a hybrid system consisting of rough-fuzzy and evolutionary computation methods.

ACKNOWLEDGMENTS

I am indebted to Professors Z. Pawlak, H. Skarzynski and A. Skowron for their encouragement and interest in this research work.

I am very thankful to Professor J. Kacprzyk who supported the idea of this new book from the beginning.

Much gratitude I owe my distinguished Teachers of Sound Engineering and Acoustics – Professors M. Sankiewicz and G. Budzyński for introducing me to these interesting, interdisciplinary scientific domains.

I thank all my colleagues and students from the Multimedia Systems Department of the Gdansk University of Technology for the discussions and support.

I would also like to thank my husband – Prof. A. Czyżewski for his encouragement and for sharing scientific interests with me.

Finally, I would like to express my gratitude to the members of the Editorial of the Springer-Verlag, in particular Shanya Rehman, Heather King Dr. Thomas Ditzinger, who supported me in a very professional way.

January, 2005
Gdańsk, Poland

Bożena Kostek

CONTENTS

1 INTRODUCTION

Over the last decade, a series of publications has brought and established new research areas related to music, and intensified the research verging on several disciplinary boundaries, typically dealt with separately. The explosion of collaboration and competition was triggered by the Internet revolution. Research achievements published in the Internet, along with audio and video available through the Internet have made research more efficient. This creates enormous possibilities and synergy. Also standards are more easily defined and implemented. On the other hand, content search of the Internet resources must in response bring new solutions to the problem – most possibly in the form of new standards and technology. Among new emerging areas are: Music Information Retrieval (MIR), Semantic Audio Analysis (SAA), music ontology, and many others. Music Information Retrieval refers to data extraction and retrieval from musical databases found on the Internet. The MIR strategic plans were defined and re-defined many times. Strong collaboration, and at the same time strong competition, afforded solutions to many problems defined within the scope of MIR, and overcame some of the largest obstacles found in this field. In addition, these problems have been addressed by technology, thus no research plans have been immune to the demands of an increasingly competitive technology environment.

There exist several definitions on semantic audio analysis. In one of them SAA means the extraction of features from audio (live or recorded) that either have some relevance to humans (e.g. rhythm, notes, phrases) or some physical correlate (e.g. musical instruments). This may be treated as complementary to human-entered metadata. In order to differentiate between human-entered metadata and semantic data, the latter issue constitutes a form of 'technical metadata', which can accompany a recording or broadcast. Thus metadata are important elements of SAA, and should cover both the extraction of features and their semantic representation. This book will highlight examples where SAA can supplement interactions with music and audio.

Human communication includes the capability of recognition. This is particularly true of auditory communication. Information retrieval can be

Bożena Kostek: *Perception-Based Data Processing in Acoustics*, Studies in Computational Intelligence (SCI) **3**, 1–5 (2005)
www.springerlink.com

investigated with cognitive systems engineering methodologies. Music information retrieval turns out to be particularly challenging, since many problems remain unsolved to this day.

Topics that should be included within the scope of the aforementioned areas include: automatic classification of musical instrument sounds and musical phrases/styles, music representation and indexing, estimating musical similarity using both perceptual and musicological criteria, recognizing music using audio and/or semantic description, building up musical databases, evaluation of MIR systems, intellectual property rights issues, user interfaces, issues related to musical styles and genres, language modeling for music, user needs and expectations, auditory scene analysis, gesture control over musical works, and many others. Some topics contained within the notion of MIR are covered by the MPEG-7 standard, which provides description of the multimedia content in order to support better interpretation of information.

It should be stressed that solving these problems requires human assistance. Many features of multimedia content description are based on perceptual phenomena and cognition. The preparation of format description, both numerical and categorical, is done on the basis of understanding the problem area. Information retrieval systems are presupposed to give an exact match to documents involving the same cues to the user query. However, operations, which are behind the query do not always provide good responses to the user's interest. This means that retrieving multimedia content on the basis of descriptors would also require human assistance. Decision systems may produce numerous rules generated in the mining process. This necessitates the provision of the generated rules for post-processing. Another problem which needs attention is the processing of unknown, missing attribute values or incomplete data when acquiring knowledge from databases. To improve information retrieval quality, various strategies were proposed and used, such as probabilistic, clustering and intelligent retrieval. The latter technique often uses concept analysis requiring semantic calculations.

The MPEG-7 standard refers to metadata information contained in the Internet archives. This notion is often applied to the value-added information created to describe and track objects, and to allow access to those information objects. In this context descriptors that are well-defined provide means for better computing and improved users interfacing and data management. It can easily be observed that these low-level descriptors are more data- than human-oriented. This is because the idea behind this standard is to have data defined and linked in such a way as to be able to use it for more effective automatic discovery, integration, and re-use in various applications. The most ambitious task is, however, to provide seamless

meaning to low- and high-level descriptors. In such a way data can be processed and shared by both systems and people.

There seems to exist a way to change primitives into higher abstraction levels, namely semantics. One of the most interesting concepts are the so-called 'computing with words' introduced by Zadeh, and the perception-based data processing which refer to the fact that humans employ words in computing and reasoning, arriving at conclusions expressed as words from premises formulated in a natural language. Computing with words can be a necessity when the available information is too imprecise to justify the use of numbers or can be a right solution when it is in better rapport with reality. It seems that this paradigm of computing can be used with success in music information retrieval, as it offers better processing of subjective descriptors of musical instrument sounds and enables the analysis of data that result in a new way of describing musical instrument sounds. An example of such processing was recently introduced by the author. It was proposed that categorical notions would be quantities partitioned by using fuzzy logic. Lately, Zadeh presented an overview of fuzzy logic defined in terms of computational rather than logical sense. In his overview he suggested that fuzzy logic has four principal aspects. The first one refers to fuzzy logic understood in narrow sense, thus it is the logic of approximate reasoning. The second aspect is related to classes that have unsharp boundaries. The third one is concerned with linguistic variables, which appear in fuzzy rules, designated for control applications and decision analysis. The fourth aspect, a so-called epistemic facet, is related to knowledge processing, meaning and linguistics. Applications related to the last mentioned aspect are based on the concept of granularity, which reflects the ability of human sensory organs and brain to process perception-based information. Existing theories, especially probability theory, do not have the capability to operate on such information, thus the development of the methodology of computing with words is considered by Zadeh to be an important event in the evolution of fuzzy logic.

It may be observed that musical object classification using learning algorithms mimics human reasoning. These algorithms constitute a way to handle uncertainties in musical data, so they are especially valuable in domains in which there is a problem of imprecision and a need for knowledge mining. Such algorithms often need human supervisory control, thus user modeling is also necessary for retrieval processes. This remark refers to both rule-based systems and neural networks in which an expert controls the algorithm settings and the choice of feature vectors.

The research studies, introduced and examined in this book, often represent hybrids of various disciplines. They apply soft computing methods to selected problems in musical acoustics and psychophysiology. These are

discussed on the basis of the research carried out in the MIR community, as well as on the results of experiments performed at the Multimedia Systems Department of Gdansk University of Technology. The topics presented in this work include automatic recognition of musical instruments and audio signals, separation of duets, processing musical data in the context of seeking for correlation between subjective terms and objective measures. The classification process is shown as a three-layer process consisting of pitch extraction, parametrization and pattern recognition. Artificial Neural Networks (ANNs) and rough set-based system are employed as decision systems and they are trained with a set of feature vectors (FVs) extracted from musical sounds recorded at the Multimedia Systems Department, and others available in the MIR community. Also, genetic algorithms were applied in musical sound classification.

This book starts with a chapter that focuses on the perceptual bases of hearing and music perception. The next chapter reviews some selected soft computing methods along with the application of these intelligent computational techniques to various problems within MIR, beginning with neural networks, rough set theory, and including evolutionary computation, and some other techniques. In addition, a review of the discretization methods which are used in rough set algorithms is given. The discretization process is aimed at replacing specific data values with interval numbers to which they belong. Within this chapter, methods of sound parametrization are also discussed. This chapter aims at presenting only the main concepts of the methods mentioned, since the details are extensively covered in a vast selection of literature. Following this, the next chapter deals with musical signal separation, its second part introduces the musical phrase analysis, while the third one is focused on metadata analysis. The Frequency Envelope Distribution (FED) algorithm is presented, which was introduced for the purpose of musical duet separation. The effectiveness checking of the FED algorithm is done on the basis of neural networks (NNs). They are tested on feature vectors (FVs) derived from musical sounds after the separation process has been performed. The experimental results are shown and discussed.

The next chapter deals with the applications of hybrid intelligent techniques to acoustics, and introduces the research, which is based on cognitive approach to acoustic signal analysis. This chapter starts with a short review of fuzzy set theory. It is followed by a presentation of acquisition of subjective test results and their processing in the context of perception. Evaluation of hearing impairment based on fuzzy-rough approach is presented within this chapter. An overview of the experiments is included, with more detailed descriptions available through some of the cited author's and her team's papers. In addition, the topic of processing of acous-

tic signals based on beamforming techniques and neural networks is presented using cognitive bases of binaural hearing. Another topic related to audio-visual correlation is a subject of the consecutive chapter. Once again, a hybrid approach is introduced to process audio-visual signals.

The last chapter outlines the concluding remarks which may be derived from the research studies carried out by the team of researchers and students of the Multimedia Systems Department, Gdańsk University of Technology. An integral part of each chapter is a list of references, which provide additional details related to the problems presented in the consecutive book sections.

2 PERCEPTUAL BASES OF HEARING AND MUSIC PERCEPTION

2.1 Perceptual Bases of Hearing Perception

2.1.1 Human Auditory System

The human auditory system pertains to the entire peripheral auditory mechanism. Classically, the peripheral auditory system has been divided into three basic parts - the outer ear, the middle ear, and the inner ear. Each part of the ear serves a specific purpose in the task of detecting and interpreting sound. The outer and middle parts form the conducting apparatus (Durrant and Lovrinic 1997; Gelfand 1998).

The outer ear serves to collect and channel sound to the middle ear. Incoming signals are collected by the auditory canal and then led to the middle ear. They cause the tympanic membrane (*eardrum*) to vibrate. In addition, the outer ear provides protection for the middle ear and prevents damage to the eardrum. Because of the length of the ear canal, it is capable of amplifying sounds with frequencies of approximately 3000 Hz. The sound pressure gain is about 10 dB. As sound travels through the outer ear, it still is in the form of a pressure wave, with an alternating pattern of high and low pressure regions. It is not until the sound reaches the eardrum at the interface of the outer and the middle ear that the energy of the mechanical wave becomes converted into vibrations of the inner bone structure of the ear.

The middle ear is an air-filled cavity which contains three tiny bones: hammer (*malleus*), anvil (*incus*) and stirrup (*stapes*), known collectively as the ossicular chain. The middle ear serves to transform the energy of a sound wave into the internal vibrations of the bone structure and to transfer these vibrations via the ossicular chain and the oval window into the inner ear. Since the pressure wave striking the large area of the eardrum is then

Bożena Kostek: *Perception-Based Data Processing in Acoustics*, Studies in Computational Intelligence (SCI) **3**, 7–37 (2005)
www.springerlink.com

concentrated on the smaller area of the stirrup, the force of the vibrating stirrup is nearly 17 times larger than that of the eardrum. Therefore the ratio of the area of the eardrum to the stapes footplate area defines the middle ear transformer. Compression forces the eardrum inward and rarefaction pushes it outward, thus making the eardrum vibrate at the frequency of the sound wave. Being connected to the hammer, the eardrum sets the hammer, the anvil, and the stirrup into motion which again is of the same frequency as the sound wave. The stirrup is connected to the inner ear so its vibrations are transmitted into the fluid of the middle ear where they create a compression wave. The three tiny bones of the middle ear act as levers to amplify the vibrations of the sound wave. Due to the mechanical characteristics of the stirrup, its displacements are greater than those of the hammer, which results in additional amplification of 1.3 times. Thus, the total amplification of the middle ear transformer of 17 x 1.3 is obtained. The middle ear does not perfectly match the impedance of the cochlea to the air, so some portion of energy is reflected. It is assumed that only 60% of the sound energy passes from the middle ear to the cochlea. Furthermore, the transfer of the sound energy through the middle ear is not constant across frequency. In addition, the middle ear is connected by the Eustachian tube to the mouth. This connection enables the equalization of pressure within the air-filled cavities of the ear (Durrant and Lovrinic 1997; Gelfand 1998).

The inner ear houses the sensory organ of hearing (cochlea) as well as the vestibular system. The latter part assists in maintaining balance. The cochlea is a snail-shaped organ which can stretch up to 32 mm approximately. It is filled with a sea water-like fluid. The inner surface of the cochlea is lined with over 17 000 hair-like nerve cells which perform one of the most critical roles in the process of hearing. There are two types of hair cells, namely inner and outer hair cells. There is a single row of inner hair cells, and typically there are three rows of outer hair cells. These nerve cells differ in length by minuscule amounts and they also show different resiliency to the fluid which passes over them. As a compressional wave moves from the interface between the hammer of the middle ear and the oval window of the inner ear through the cochlea, the small hair-like nerve cells are set in motion. Each hair cell has a natural sensitivity to a particular frequency of vibration. When the frequency of the compressional wave matches the natural frequency of a nerve cell, that nerve cell resonates with a larger amplitude of vibration. The increased vibrational amplitude induces the cell to release an electrical impulse which passes along the auditory nerve towards the brain where it is decoded and interpreted. Only about 5% of neurons connect to the outer hair cells, this means that each neuron receives input from numerous outer hair cells. Their activity is

summated by the neurons to improve sensitivity. The other 95% of neurons connect to the inner hair cells providing better discrimination.

Since the basilar membrane of the cochlea, on which the travelling waves appears, has a variable spatial sensitivity to specific stimulation frequencies, therefore it works like a frequency-place processor. There is also a simple relation between the place and the sensitivity of neurons to a specific characteristic frequency referred to as 'tonotopical organization'. Both Fletcher and Zwicker independently drew a simple conclusion that the auditory system can be modeled with a set of band-pass filters located on frequency axis reflecting critical bands. Moore made the statement that the shape of critical bands can be approximated as rectangular which makes them different from the shape of hearing filter characteristics on the cochlea membrane. He suggested the scale of equivalent rectangular bands expressed in ERB units. The auditory system is characterized with exponential curve easily approximated by the filter, whose impulse response has the gamma filter envelope modulated by medium frequency (Durrant and Lovrinic 1997; Gelfand 1998).

2.1.2 Auditory Perception

Several investigations (Zwicker and Feldkeller 1967; Zwicker and Fastl 1990) have shown that many aspects of the human auditory perception are in fact almost independent of individual preferences. The most important ones are the occurrence of masking, the perception of loudness, and the perception of pitch. These characteristics of the human auditory system can be modeled in order to get an objective estimate of perceived audio quality (Thiede 1999). Most of these characteristics can be approximated by analytical expressions which, for example, have been proposed in the works of Terhardt (Terhardt 1979; Zwicker and Terhardt 1980; Terhardt 1992).

The listener-independent characteristics of the auditory perception form a low-level model of the human auditory system. Besides the study of Zwicker, the works of Moore (Moore 1989, 1996) also include descriptions of most aspects of the auditory perception but they form a slightly different model. Even though the results of the experiments carried out by Moore (1989, 1996) are often considered to correspond better to the physiological structure of the auditory system, the model proposed by Zwicker has proven to work rather well when applied to perceptual coding and perceptual measurement (Thiede 1999).

The subjectively perceived loudness of an audio signal does not solely depend on its sound pressure level but it also depends on other signal

characteristics such as its frequency distribution for example. This effect is taken into account by replacing the decibel scale with the *phon scale*. The loudness of a signal given in *phons* corresponds to the sound pressure level in decibels of a pure tone at 1 kHz which produces the same perceived loudness as the measured signal.

When the sound pressure level of a pure tone that produces a certain loudness is plotted as a function of frequency, the resulting curves are called the *equal loudness contours*. The equal loudness contours for a sound pressure level of 0 dB is of particular interest, as it corresponds to the *absolute threshold* of hearing (see Fig. 2.1). Terhard proposed to approximate this threshold curve by the following expression:

$$LT_q = 3.64 \cdot f^{-0.8} - 6.5 \cdot \exp[-0.6 \cdot (f - 3.3)^2] + 10^{-3} \cdot f^4 \qquad (2.1)$$

This function is used in audio coding strategies.

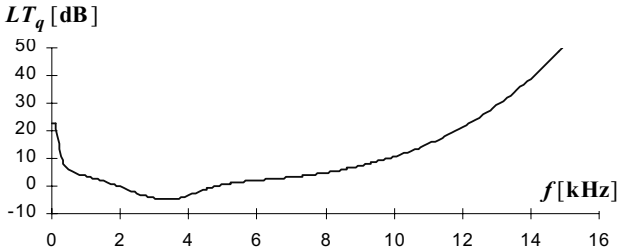


Fig. 2.1. Hearing threshold approximation

Before the detection of a sound takes place on the basilar membrane, the acoustical signal is transferred via the ear canal and the middle ear to the inner ear. The transfer function can be described by the following expression (Thiede et al 2000):

$$A(f) = -0.6 \cdot 3.64 \cdot f^{-0.8} + 6.5 \cdot \exp[-0.6 \cdot (f - 3.3)^2] - 10^{-3} \cdot f^4 \qquad (2.2)$$

or by its simplified form given by Moore (1996):

$$A(f) = 6.5 \cdot \exp[-0.6 \cdot (f - 3.3)^2] - 10^{-3} \cdot f^4 \qquad (2.3)$$

where f is given in kHz. A small difference between these two functions may be observed for low frequency range.

Zwicker and Terhard (1980) proposed an expression for the critical bandwidth:

$$\Delta f = 25 + 75 \cdot [1 + 1.4 \cdot f_c^2]^{0.69} \tag{2.4}$$

where the center frequency of the band f_c is denoted in kHz, and the bandwidth Δf in Hz.

The critical band rate is expressed in Bark units and is usually referred to by the symbol of *z*. Schroeder (Schroeder et al 1979) proposed a very simple approximation to calculate the frequency that corresponds to a given critical band rate:

$$f = 650 \cdot \sinh(\frac{b}{7}) \text{ or } b = 7 \cdot \operatorname{arcsinh}\left(\frac{b}{650}\right) \tag{2.5}$$

The model given above was designed for a frequency range relevant to speech, and is consistent with psychoacoustical measurements of up to 5 kHz which corresponds to 20 Barks. On the other hand, Terhard and Zwicker (Zölzer 1996) proposed another, more precise formula which may be applied to the entire frequency band:

$$b = 13 \cdot \operatorname{arctg}(0.76 \cdot f) + 3.5 \cdot \operatorname{arctg}\left[(\frac{f}{7.5})^2\right] \tag{2.6}$$

where b and f are frequencies denoted correspondingly in Bark and kHz. This expression corresponds well to experimental data used by Zwicker.

The formula given below is still another expression describing the relationship between the Hz and the Bark scales. It was introduced by Tsoukalas et al in 1997:

$$b = \frac{26.81 \cdot \omega}{1960 + \omega} - 0.53 \tag{2.7}$$

where ω is given by the following expression: $\omega = 2\pi f$.

Modeling the Auditory System

It is worth remembering that for the signal perceptual analysis various types of filter bank techniques may be used. Among them the most often applied are the one-third-octave band filters, the Bark filters, the Gammatone filter banks, and warped filters (Thiede 1999).

The filter bandwidths of the one-third-octave band filters are proportional to center frequencies and thus correspond to the auditory frequency

scale in the upper frequency range. The Bark scale can be approximated by combining neighboring filters at low center frequencies. This technique is used in the loudness measurement scheme introduced by Zwicker. However, the one-third-octave filters yield only a rough approximation of the auditory filter shapes, and their approximately rectangular shapes are not always desirable in auditory modeling. A measurement method based on the so-called BARK-transform was first published by Kapust (1989). It originally used multiple FFTs with different window functions to approximate the spectral and temporal resolution of the human auditory filters. The frequency-to-pitch mapping is therefore approximated by a staircase function. It uses 11 different window functions for a decomposition into 25 frequency bands. The Gammatone filter bank corresponds to an extension of the FTT (Fourier Time Transform) to filter orders higher than one. In many publications, the fourth order Gammatone filter bank is considered to be a good approximation for the auditory filter functions. However, this assumption holds for a limited dynamic range of approximately 50 dB (Thiede 1999).

All the above-mentioned models work on a low level of abstraction and do not explain the human ability of drawing high-level conclusions concerning such information as:

- Musical information: melody, rhythm, metrum, harmony, modality;
- Environmental information: spatial location, rooms characteristics, background sounds identification;
- Cognitive information: instrument identification, recognition of a melody, a composer or a musical style;
- Emotional information: identification of emotional intentions of a composer or a musician.

Other classes of psychoacoustic models can also be found in publications concerning the application of auditory models to signal processing. For example, for the sound separation problem two main pattern recognition models and time-domain models where introduced. In addition, these models are related to theories on pitch perception of complex tones published by Goldstein in 1973, Wightman in 1982 and Terhardt in 1979, and also to some more recent theories included in works published by Meddis and Hewitt in 1991 or Cohen et al. in 1995. The latter one is based on two levels of analysis: spectral analysis and the main processing module, i.e. pattern recognition module. The most popular model of such type is Goldstein's optimum processor (Goldstein 1973) which first detects the maxima of the spectrum for each ear, and then – assuming that all detected components are noisy representatives of harmonic frequencies – calculates the estimate of the greatest probability of the fundamental fre-

quency. The model can explain the process of creating the missing harmonic components and the dychotomic phenomenon (when each ear is stimulated with a set of different harmonic components). Other popular models include: Schroeder's histogram which is a detector of a fundamental frequency operating through the search of all integer multiples of spectrum maxima, and Parsons' model based on a similar approach and used in speech acoustics based on a similar approach and used in speech acoustics (Walmsley 2000).

Time-domain psychoacoustic models work differently. They assume that in the process of sound event perception, time information is significant. The nerves participating in hearing 'set off' at the same moments of the stimulus activity provided that the frequency of its appearance is lower than 5 kHz. Therefore each neuron transfers the information about the periodic character resulting from the time they were stimulated, additionally fulfilling the conditions resulting from the above-mentioned tonotopical organization of hearing. The models initially implement the filtration simulating the operations of the cochlea (Zwicker 1961, Moorer 1997), and then periodic detection is introduced. The most popular method currently involves the hybrid time-domain and place-domain models, like Patterson and Holdsworth's (1996) or Meddis and Hewitt's (1991).

The model includes an initial filter simulating the operations of external and internal ear, behind which the set of 128 overlapping filters working in critical bands is placed, aimed at simulating the response of the stimulated basilar membrane of the cochlea. Separate signals are sent to simulate the activation of hair cells, while the resulting impulses from neurons undergo autocorrelation analysis. Data received from each channel are then averaged, which results in a correlogram allowing for recovering an appropriate sound pitch. The autocorrelation analysis exploits periodicity in the cochleogram.

The conditions of Patterson and Hewitt's model make it a rather functional than strictly physiological model. The level of cochlea filtration uses the set of above-mentioned filters based on the gamma function and located on appropriate equivalent rectangular bands, while the level of nerve detection threshold uses logarithmic compression and adaptation level.

2.1.3 Masking

The limited spectral and temporal resolution of the ear in combination with a limited dynamic range produces a phenomenon called masking. Masking is a process by which the threshold of audibility of one sound is elevated,

causing sensitivity decrease in the presence of another sound. When two signals are sufficiently close to each other in time and frequency, the weaker signal may become inaudible due to the presence of the stronger one. The signal component that is masked is called a *maskee* and the signal component that masks another one is called a *masker*. Typically masking is explained separately in the frequency and in the time domains. The simultaneous masking refers to the case when a masker and a maskee are presented at the same time, and they are close in frequency domain in terms of critical bands. If masking depends mainly on the location in the time domain, i. e. a masker and a maskee have a similar spectral shape, it is called temporal masking (Durrant and Lovrinic 1997; Gelfand 1998; Pohlmann 1995).

Simultaneous Masking

The effect of simultaneous masking is interpreted as a result of the spread of neural excitations from the basilar area that corresponds to the frequency range of the sound stimulus, into the areas that actually should not respond to it. This is modeled by assigning energy within each critical band**:** an excitation in the same band, and some excitations in the adjacent critical bands. The excitations assigned to the adjacent critical bands are determined by the shape of the masking curves. This results in numerous excitations for each critical band (one that originates from the energy in the same critical band and several others that originate from the energies in the adjacent bands). The excitation is determined by the maximum of the partial excitations, even though this does not correspond to the additivity (Thiede 1999). In Fig. 2.2 four masking curves are shown (tones are masked by narrow band noise) at 60dB. In Fig. 2.2 masking thresholds are plotted with logarithmic frequency. Masking curves on the Bark scale are presented in Fig. 2.3. Without a masker, a signal is inaudible if its SPL is below the threshold of the quiet (dashed curve) which depends on frequency and covers a dynamic range of more than 60 dB as shown in Figs. 2.2 and 2.3. The masking threshold depends on the sound pressure level (SPL) and the frequency of a masker, and on the characteristics of a masker and a maskee. Simultaneous masking curves are asymmetrical. The slope of the masking threshold is steeper towards lower frequencies, i.e., higher frequencies are more easily masked. In addition, in Fig. 2.4 an approximation of a masking curve for a single tone is presented.

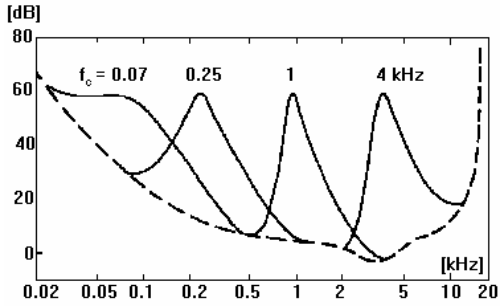


Fig. 2.2. Masking curves at 60 dB on logarithmic scale (Zwicker and Zwicker 1991); *y*-axis refers to threshold, excitation level

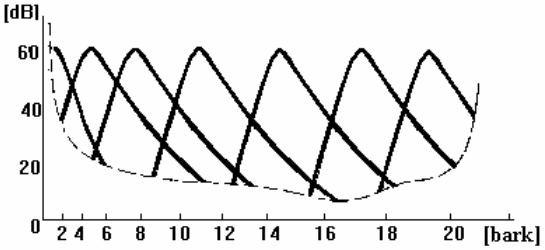


Fig. 2.3. Masking curves on a bark scale; *y*-axis refers to threshold, excitation level

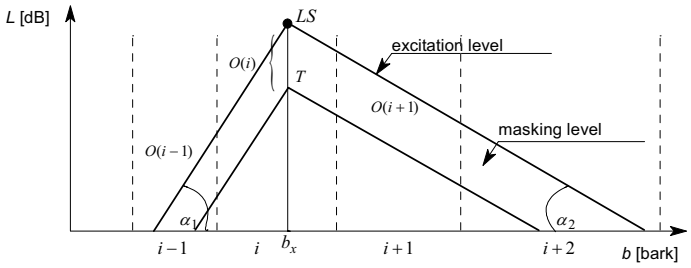


Fig. 2.4. Approximation of a masking curve for a single tone

Let $s_1 = \mathrm{tg}\alpha_1$ and $s_2 = \mathrm{tg}\alpha_2$, then the slope of the approximation function of the basilar membrane response when excited by a signal *LS* [dB] of the frequency of f_x [kHz] expressed as dB/bark, yields (Beerends and Stemerdink 1992):

$$\begin{cases} s_1 = 31 \\ s_2 = 22 + \min(0{,}23 \cdot f_x^{-1}, 10) - 0{,}2 \cdot LS \end{cases} \tag{2.8}$$

or using center frequency $f_c(i)$ of the *i*th critical band, this expression can be presented as (Zölzer 1996):

$$\begin{cases} s_1 = 27 \\ s_2 = 24 + 0{,}23 \cdot f_c^{-1}(i) - 0{,}2 \cdot LS \end{cases} \tag{2.9}$$

If the source signal consists of many simultaneous maskers, a global masking threshold should be computed that describes the threshold of just noticeable distortions as a function of frequency.

Temporal Masking

Temporal masking phenomenon refers to masking effects between two sounds which are not produced simultaneously. Temporal masking is schematically presented in Fig. 2.5. As illustrated in Fig. 2.5, the amount of masking depends on the relative intensities of the two sounds, their frequencies and the time interval between them. In temporal masking two phases may be discerned: premasking (*backward masking*) and postmasking (*forward masking*). In the case of premasking, signal components are masked before the onset of the masker, and in the case of postmasking, signal components are masked after the termination of the masker. The first effect may take place in 15-30ms before the actual sound starts, whereas the duration of the second one is up to 250ms. The premasking phase implies that a loud signal can mask another signal before the former one is actually present. It is usually explained by the assumption that loud signals are processed faster than weak signals and may therefore overtake the maskee during the processing of the signal, either on the auditory nerve or later on in the higher levels of the auditory system. Temporal interactions do not always produce reduced sensitivity. This may be due to the so-called sensitization, the occurrence of a preceding sound, or the adaptation phenomena. The latter is related to a decreased responsiveness resulting from sustained stimulation (Durrant and Lovrinic 1997; Gelfand 1998; Thiede 1999).

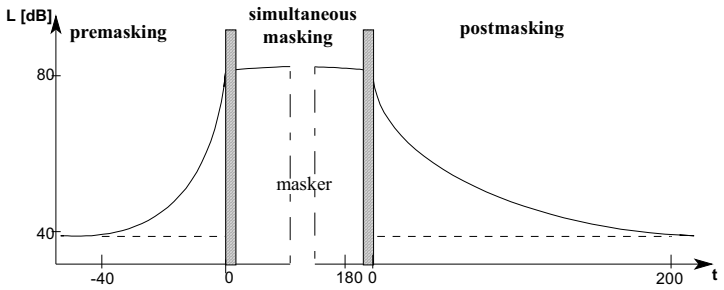


Fig. 2.5. Temporal masking

The postmasking effect is often modeled using exponential function, i.e.:

$$L_p = K \cdot \left(e^{-t/\lambda_1} + e^{-t/\lambda_2}\right) \cdot L_m \tag{2.10}$$

where L_p denotes masking level, L_m is the masker level, and λ_1, λ_2 are time constants.

Another concept was presented by Jesteadt et al., who on the basis of psychoacoustic tests, concluded that the *amount of masking M* is a function of frequency and a time gap between a *masker* and a *maskee* signal (Jesteadt et al 1982). The function is given below.

$$M = a \cdot \left(b - \log_{10} \Delta t\right) \cdot \left(L_m - c\right) \tag{2.11}$$

where Δt is a time gap (in ms) between a *masker* and a *maskee*, the level of which is denoted as L_m in dB SL. Parameters *a*, *b*, *c* are set experimentally. The factor $a \cdot \left(b - \log_{10} \Delta t\right)$ denotes the slope of the masking curve with regard to Δt.

It should be noticed that Zwicker defined the postmasking effect in terms of the following function (Kapust 1992):

$$D = 1 - \frac{1}{1{,}35} \operatorname{arctg}\left(\frac{t}{13{,}2 \cdot d^{-0{,}25}}\right) \tag{2.12}$$

where *D* denotes masking effect duration, and *d* is the duration of the masker.

The level difference between a signal and the maximum of the masking threshold it produces is called *masking index* (Zwicker and Fastl 1990). It depends on the center frequency of the masker but is assumed to be independent of the signal level. The linear representation of the masking index is called *threshold factor* (Zwicker and Fastl 1990).

In practical experiments the additivity of masking thresholds should be assumed. If masking is produced by several signal components located at different positions in the time-frequency plane, the most obvious assumption would be that the overall masking threshold is either given by the sum or by the maximum of the masking thresholds produced by each of the signal individual components. In fact, the masking threshold produced by a multi-component signal is much higher than the sum of the thresholds produced by its individual components.

Let A and B be two masking signals of level x_A and x_B, located at different location in time, expressed on a linear scale. In addition, let T_A and T_B be masking thresholds for instance t_M correspondingly for signals A and B, then the dependence between the thresholds produced by the individual signal components and the threshold produced by the complete signal $T_{A,B}$ can be approximated by (Penner 1980; Penner and Shiffrin 1980):

$$T_{A,B} > T_A + T_B \tag{2.13}$$

The difference $T_{A,B} - (T_A + T_B)$ is called additional masking. For expressing additivity of masking thresholds function J was introduced.

$$J(T_{A,B}) = J(T_A) + J(T_B) \tag{2.14}$$

which may be generalized for multi-component signal excitation:

$$J(T_{M*}) = J(T_{M1}) + J(T_{M2}) + \ldots + J(T_{MI}) = \sum_{i=1}^{I} J(T_{Mi}) \tag{2.15}$$

Computation of function J is performed on the basis of the model proposed by Penner (Penner 1980; Penner and Shiffrin 1980). An auditory system denoted as $y(t)$ for a time instance t, excited by the signal $x(t)$ may be expressed as:

$$y(t) = g\left\{ \int_{-\infty}^{t} u(t, \tau, x(\tau)) \mathrm{d}\tau \right\} \tag{2.16}$$

where u refers to excitation pattern on the basilar membrane, and g refers to transfer function for the auditory system. After several additional assumptions, function J, expressed in terms of the auditory system, is given by:

$$J(T_M) = \int_{-\infty}^{t_M} u[t_M, \tau, x_M(\tau)]\,\mathrm{d}\tau \tag{2.17}$$

where t_M is an instance for which the masking threshold T_M is expressed. Typically, values of particular coefficients and factors are set experimentally.

2.2 Perceptual Bases of Music Perception

2.2.1 Music Perception

Perception is by definition the act of perceiving, cognizance by the senses or intellect, apprehension by a bodily organ or by mind of what is presented to them. Another definition lists several expressions synonymous with perception, namely: becoming aware of something via the senses, the process of perceiving, knowledge gained by perceiving, a way of conceiving something, the representation of what is perceived. Thus it is a conscious mental awareness and interpretation of a sensory stimulus.

Music perception is an interdisciplinary area which combines a number of disciplines, such as physics, psychoacoustics, mathematics and musicology. Each of them plays an equally significant role in the understanding of musical phenomena. The target of physics is understanding the mechanism of sound creation in musical instruments which became the core of design of new and improved instruments. Psychoacoustics focus on the other side of sound nature – effects of music upon humans which in turn approaches the area of cognitive psychology, dealing with the highest level of organization of the heard sound and the area of the sound scene analysis. Both the musical theory and the musical psychology have had a significant impact on this area. They focus on high-level modeling of musical structures, dealing with such structures as key, metrum, or harmony.

Pitch is the attribute by which sounds are ordered along the frequency axis from low to high. Low pitches are associated with low frequencies, and conversely sound of high pitches – with high frequencies. Frequency is not the sole determinant of pitch. Changes in intensity can also cause difference in pitch perception. The ear is more sensitive to frequency changes at higher frequencies. The pitch of complex sounds is based on the periodicity phenomenon. It is believed that pitch processing is almost synonymous with periodicity processing. Pitch perception, is largely insensitive to

the phase of individual partials. It may be said that the human hearing system possesses an inaccurate periodicity detector that is used for determining pitch. The evidence for this claim may be enforced by the following facts:

- The auditory system is able to perceive several partials of a complex tone separately.
- It is assumed that the auditory system is able to recognize and memorize the frequency intervals existing between those partials. Certain periodic signals having various fundamental frequencies are of highest biological relevance and occur often; the voiced sounds of speech. Hence, the chance is high that the auditory system will memorize that specific harmonic arrangement of partials as a template.
- More generally, when a higher sensory system has memorized a specific stimulus arrangement, this particular arrangement, if presented again, will be perceived as a distinct entity rather than as a meaningless complex.
- When a higher sensory system has developed the ability to perceive a pattern as an entity, the identification of that pattern will not break down when some of its elements are absent, provided that the remaining elements are objectively sufficient to identify the pattern.
- Another way to observe the pitch periodicity is to present a group of pure tones together, equally spaced in frequency, then a fundamental pitch will be detected which corresponds to the common frequency difference between individual components.
- In addition, the inability of discrete bands of noise centered around the fundamental frequency to mask out or obscure the residue, meant as pitch sensation.

Based on a simple model of the auditory system one cannot explain a lot of phenomena, e.g. complex tone perception or an even less comprehensible phenomenon of missing fundamental frequency perception. In the theory of virtual sound pitch proposed by Terhardt hearing perception is based on the combination of analytical hearing and holistic perception. The approach seems appropriate in cases of perception of such instruments as bells, whose spectrum is non-harmonic, and the tone heard is specified by the components from fourth to sixths.

What human hearing is lacking in certain areas (low sensitivity, masking effects) is made up for in speed, accuracy and the exceptional ability of high-level conclusion-drawing. The ear is able to isolate a separate speaker from the crowd (so-called “cocktail party” effect) and to track a selected voice in a polyphonic recording even without the help of other senses. Psychoacoustics and psychology experts are interested in understanding

the physical, nerve, and psychological processes which enable sound perception, in order to use the information to create the models of the auditory system. One can focus either on the low level modeling in order to capture the essence of the auditory system, or on perceptual models created on a higher level of the structure.

Other physiological aspects of music are as follows (Allott 2004):

- Synaesthesia

It is hypothesized that synaesthesia is possible due to some extra connections in the brain which occur between areas concerned with auditory and visual perceptions. It is referred to as "seeing" sounds, "hearing" colors, that results from mutual relations and connections of different senses. The most common musical synaesthetic experience is seeing colors or patterns when music is heard or composed.

- Temporal (rhythmic) effects

It is assumed that the perception of musical rhythm is crucial to understanding the auditory perception abilities of the right cerebral hemisphere. Apart from the rhythm of breathing, the other dominant rhythmic sound in our lives is the heartbeat (McLaughlin 1970).

- Motor effects

Music has a direct relation to the nervous organization of postures and movements. Composition, performance and listening imply wide involvement of the cerebral motor cortex, subcortical motor and sensory nuclei and the limbic system (Critchley et al 1997).

- Other body-based responses

Perceptual and emotional musical experiences lead to changes in blood pressure, pulse rate, respiration, and other autonomic functions. These autonomic changes represent the vegetative reflections of psychological processes. During the act of conducting, the highest pulse frequencies are not reached at moments of greatest physical effort but at passages producing the greatest emotional response, passages which the famous conductor von Karajan singled out as being the ones he found most profoundly touching.

- Neural patterning

There are strong analogies or structural similarities between music and the fundamental activities of the nervous system. The characteristics of nerve impulses - timing, intensity, synchronicity, frequency-contrasts, patterning generally - can be set in parallel with many aspects of musical construction. The fiber tracts of the auditory system are arranged tonotopi-

cally, so that the frequency organization of the cochlea is maintained at all levels from the auditory nerve to the cortex.

A quiet noise produces only a few pulses per second, a loud one several hundred per second. The pulses are identical, whatever the intensity of the stimulus, in every nerve (McLaughlin 1970).

The implications for the perception of music of this identity of nerve impulses and patterning are discussed by McLaughlin. When the incoming signal, as in the case of music, is a pattern which has no immediate function as a useful sense impression, equivalent patterns from other sense modes will be activated. The selection and succession of the musical notes may have no significance for us but the electrical patterns into which they are translated can be compared and identified with patterns from other sources (Allott 2004).

- Emotions

The effects of music are very complex and constitute a synthesis of many emotions and feelings. There are also other qualitative aspects of sound that appear to reflect psychological judgments about sound (Allott 2004).

- Haptic processing

In her mini-tutorial Lederman discussed psychophysics as a field of experimental psychology that uses specific behavioral methods to determine the relationship between the physical world and humans' subjective experience of that world, the so-called human haptic processing. Experiments that are conducted in this domain are specifically designed to discover which physical parameters determine a subjective perceptual dimension. Especially important is to evaluate humans' sensation in terms of intensity, spatial and temporal variations in mechanical, kinesthetic, and thermal inputs (Lederman and Klatzky 1996; Lederman 2004).

The fields of *sonification* and *auditory display*, though, have provided an impressive body of research, especially since the mid 1990s. Understanding of the audiovisual, multi-modal, and dynamic become aspects of new multi-linear media.

Musical gestures as suggested by Kurth and later by Scruton are isomorphic with expressive motion (Kurth 1991; Scruton 1979). Such notions as sonic embodiments of image schemas, or ionic components in music semiotic appeared in the literature published by Lidov in 1999 and Hatten in 1977-2002. It is assumed that the relationship between music and motion is fundamental to music processing. Eitan and Granot concluded that musical parameters affect motion imagery strongly and in a diverse way. Moreover, it is possible to associate specific musical and motional parame-

ters. In addition, intensity direction often matches musical and motional parameters. They also observed and tried to explain multiple mapping strategies, and found that musical-kinetic analogies are often directionally asymmetrical (Eitan and Granot 2004). From such studies some implications for music theory arose. Cognitive mapping of music into space and motion is very complex. Such finding may apply to models of pitch space, such as Larson's (Larson 1997), Schenker's, and others.

It is worth reviewing the issue of the IEEE Proceedings dealing with subjects on Engineering and Music – Supervisory Control and Auditory Communication. Especially valuable may be papers by Johannsen (2004), Canazza et al. (2004), Suzuki and Hashimoto (2004), and others of the same issue (Johannsen 2004). The domain of "Human Supervision and Control", as suggested by Johannsen, can be analyzed in the engineering sciences and in musicology from different cultural, social, and intellectual perspectives. It embraces the human activities of guidance, monitoring, and control (manual and supervisory control or sensorimotor and cognitive control) and also includes perception, information processing and action. *Supervisory Control* is methodically the most important sub-domain of "Human Supervision and Control." Other sub-domains supplement this with respect to different aspects of the creation and transfer of information, such as gestural control, motion and sound control, information retrieval, composition and analysis, sound design and multimedia, virtual environment, performance and interpretation, as well as visual, auditory, and haptic supervision and communication (Johannsen 2004).

2.2.2 Musical Instrument Sounds

Musical sounds are an important and natural means of human communication and culture. During many epochs, much effort has been aimed at creating and developing various instruments used in music. Most musical instruments generate sound waves by means of vibrating strings or air columns. In order to describe the features of musical instruments, one must first decide on a division of instruments into categories (groups) and subcategories (subgroups), aimed at pointing out similarities and differences between instruments. There are various criteria to make this separation possible, however it is often sufficient to limit this problem to only two criteria, namely the way an instrument produces sound and whether or not an instrument is based on Western musical notation (The New Grove 1980). An example of such a division of musical instruments is shown in Table 2.1. Instruments included in this table are found in most of the contemporary symphony orchestras.

Table 2.1. Division of musical instruments into categories

Category	Subcategory	Contemporary symphony orchestra musical instruments (examples)
String (or chordophone)	Bow-string	violin, viola, cello, contrabass
	Plucked	harp, guitar, mandolin
	Keyboard	piano, clavecin, clavichord
Wind (or aerophone)	Woodwind	flute, piccolo, oboe, English horn, clarinet, bassoon, contra bassoon
	Brass	trumpet, French horn, trombone, tuba
	Keyboard	pipe organ, accordion
Percussion (or idiophone & membranophone)	Determined sound pitch	timpani, celesta, bells, tubular bells, vibraphone, xylophone, marimba
	Undetermined sound pitch	drum set, cymbals, triangle, gong, castanets

With regard to the above given assumptions, the main acoustic features of musical instruments include:

- musical scale,
- dynamics,
- timbre of sound,
- time envelope of the sound,
- sound radiation characteristics.

The musical scale is a set of sounds that an instrument is capable of producing. Dynamics defines all phenomena related to the intensity of sounds. The dynamic range can be described as the relation between the level of a sound measured during the *forte fortissimo* passages and the level of a sound measured during the *piano pianissimo* passages of the composition. The dynamic range depends on the technique of playing (*musical articulation*) and it is different for continuous play (*legato*) and for single tones. This is illustrated in Fig. 2.6 (Meyer 1993). In general, string instruments are only slightly quieter than woodwind instruments and are about 10dB quieter than brass instruments.

Sound timbre is a feature that makes it possible to distinguish sound of various instruments. First of all it depends on the number, type and intensity of the component harmonics. Sounds that have few harmonics have a soft but dark sound, and those with a lot of harmonics – especially with a prevailing number of high components – have a bright and sometimes even sharp sound. The timbre is also closely correlated with the shape of the time envelope and the pitch of the sound. Sound pitch can be expressed as an absolute or relative value. The absolute representation is characterized by exact definition of a reference point (e.g. the C1 sound). In the case of relative representation the reference point is being updated all the time.

Here the reference point may be e.g. the previous sound, a sound of the previously accented time part or a sound of the time start. Pitch enables a distinction between the sound registers of an instrument. The influence of dynamics on timbre can also be observed. For string instruments, this influence is only minor because the components of more than 3kHz rise only by 1.1dB when the level of dynamics rises by 1dB. For woodwind instruments the level of these components rises by about 1.2-2.0dB and for brass instruments they can rise by as much as 3dB. An additional factor having influence on the instrument timbre is the technique of the performance, i.e. vibrato, pizzicato, martele, spiccato, etc. Higher harmonic components of brass instruments and of the flute, when played with vibrato, undergo amplitude modulation which leads to an audible change of both the dynamics and the timbre of the tone.

As mentioned before, timbre is defined as the quality which distinguishes two sounds with the same pitch, loudness and duration. The notion of musical sound timbre refers to the works done by Grey (1977) and later by Krimphoff et al. (1994), McAdams and Winsberg (1999), Wessel (1979), Reuter (1996) and many others (De Bruijn 1978; Cook 1999; Cosi et al 1984; Iverson and Krumhansl 1993; Jensen 2001; Pollard and Jansson 1982; De Poli et al 1991; De Poli et al 2001; Pratt and Doak 1976; Pratt and Bowsher 1978; Vercoe et al 1998). Three dimensions recognized by Grey were discussed in some papers and resulted in diminishing the timbral space to two dimensions. In the original paper by Grey spectral energy distribution, the presence of synchronicity in sound attacks and decays (spectral fluctuation), and low-amplitude, high-frequency energy in the initial attack represented the perceptual relationship. In other studies the first dimension is identified with the log attack time, the second one with the harmonic spectral centroid (Iverson and Krumhansl 1993) and the third one – with a temporal centroid. Wessel's similarity judgment tests show that two dimensions are related to centroid of the tone spectral energy distribution, and the velocity of the attack (Wessel 1979). Iverson and Krumhansl pointed out that both spectral and temporal attributes were important for the instrument similarity measurements. It may be seen that according to most papers dealing with perceptual space, researchers tried to limit this space to three or two parameters. It is obvious that in such a case dimensions can be easily interpreted and presented as a two- or three-dimensional projection, however in the author's opinion derived from the results of processing multidimensional feature vectors describing musical sound characteristics there is no need to limit the perceptual space to such a small number of parameters (Herrera et al 2000; Kostek 1999, 2003, 2004; Kostek and Czyzewski 2001).

Time envelope is also of importance when analyzing musical sounds. This issue will be referred to later on while discussing time-related parameters.

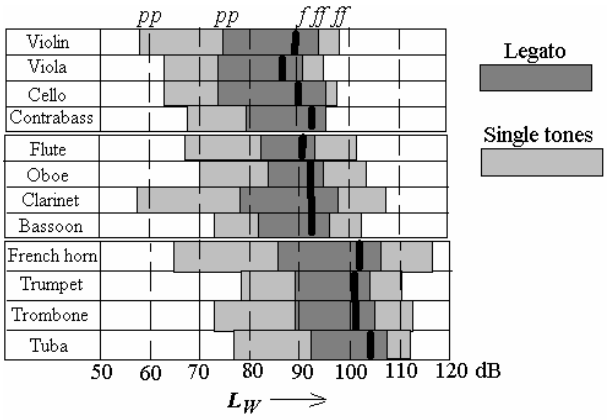


Fig. 2.6. Dynamic ranges of some chosen musical instruments ($L_W \rightarrow$ acoustic power level with reference to 10^{-12} W/m^2 (Meyer 1993))

The last feature to be mentioned here is the sound radiation characteristics. This feature depends greatly on the sound-radiating elements of a musical instrument. Although low-frequency sounds (below 500Hz) from most instruments radiate in all directions, higher-frequency components are increasingly direction-dependent. This feature creates some difficulties, especially while recording single sounds generated by a particular musical instrument.

2.2.3 Musicological Analysis

One of the most remarkable properties of the human auditory system is its ability to extract a pitch from complex tones. This is an analytical mode of listening. On the other hand, a person may also generalize what he/she is listening to. When confronted with a series of sounds, instead of hearing each sound as a unique event, he/she may choose to hear a melody pattern. Such a generalization refers to the so-called holistic mode of listening. The first approach may conclude in the recognition of an individual instrument, while the second may be thought of as the ability to recognize a musical piece, belonging to a given musical style. Musical style is a term denoting the mode of expression, or even more precisely, the manner in which a work of art is executed. It may be used to denote the musical characteris-

tics of individual composers, periods, geographical areas, societies or even social functions. From the aesthetic point of view, a musical style concerns the surface of the appearance of music (The New Grove 1980). In musicology, there are different approaches to music analysis. Schenker's approach to harmonic analysis, in relation to viewing tonal music in its simplest form, remains one of the core teachings in music theory. Schenker's theory of tonal music defines a melodic structure as a diatonic line derived by analytical reduction when the upper structure is removed. This fundamental melodic structure is called the *Urlinie*. Schenker extended this concept to fundamental composition and finally to a general concept of structural layers: background, middleground and foreground. The background reduces the music to its most significant material, often consisting of only 3-5 pitches. On the other hand, the foreground is a notated representation of the majority of all the notes in the piece in a very detailed fashion, reducing the piece to its minute elements. The general concept of structural levels provides for a hierarchical differentiation of musical components which in turn establishes a basis for describing and interpreting relations among the elements of any composition. These considerations are founded on the concept that the removal of the upper-layers constitutes the core of the musical phrase, in some cases just a single note (The New Grove 1980). This style of analysis has its roots in Gestalt theory. According to the Gestalt theory, individuals react to meaningful wholes; and therefore, learning is based on the organization of the ideas that are important, discarding less important material.

On the other hand, functional theory, described by Riemann, defines the relationships of chords to the tonic as center (The New Grove 1980). Riemann's main interest dealt with the classification of rhythmic motifs as on-stressed, interior-stressed and off-stressed, depending whether their accent fell at the beginning, in the middle or at the end. His view point was that an increase in the frequency of interior- and off-stressed rhythms brings an increase in energy. Additionally, he defined a 'rhythmic motif' as the smallest constructional unit of significant content and definite expressive value. A motif, being the fundamental unit, is at the same time a unit of energy. If two individual units are a succession of notes and they are adjacent to each other, then they are combined into a larger form. Such a form then creates a higher formation which is next in the hierarchy. Riemann's theory aims at searching for points which divide music into units. In Fig. 2.7, such a division is shown. Here, an eight-bar module is presented as 2/4 units. A two-bar module is a combination of two motifs which form a half-phrase (The New Grove 1980).

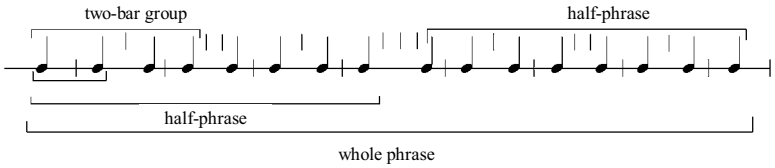


Fig. 2.7. Application of Riemann's theory to music division

Leach and Fitch proposed another approach to music analysis. The basic assumption of this method is that music is an association of musical phrase repetitions. Fragments of music that are repeated are later combined into groups. In Fig. 2.8, an example of such an analysis is shown. In the example, at the beginning a three note group is repeated three times, the tenth note is then different and therefore does not belong to a larger formation, but the next sequences of notes are also repetitions, hence they are grouped. The whole process forms a hierarchical tree-like structure (Leach and Fitch 1995).

Another theory of music analysis is based on the notion that the starting point for the analysis is the construction of a model that reflects the listener's perception of music. Such a model of structural hearing was first introduced by Lerdhal and Jackendoff, reexamined by Narmour, and again revisited by Widmer (1995). Structural hearing, according to Lerdhal and Jackendoff, is a concept that allows one to understand music as complex structures consisting of various dimensions (see Fig. 2.9).

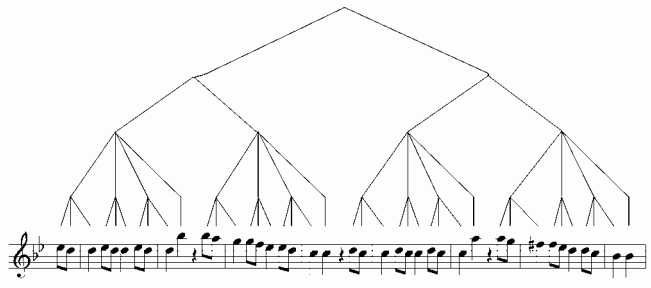


Fig. 2.8. An example of music analysis proposed by Leach and Fitch

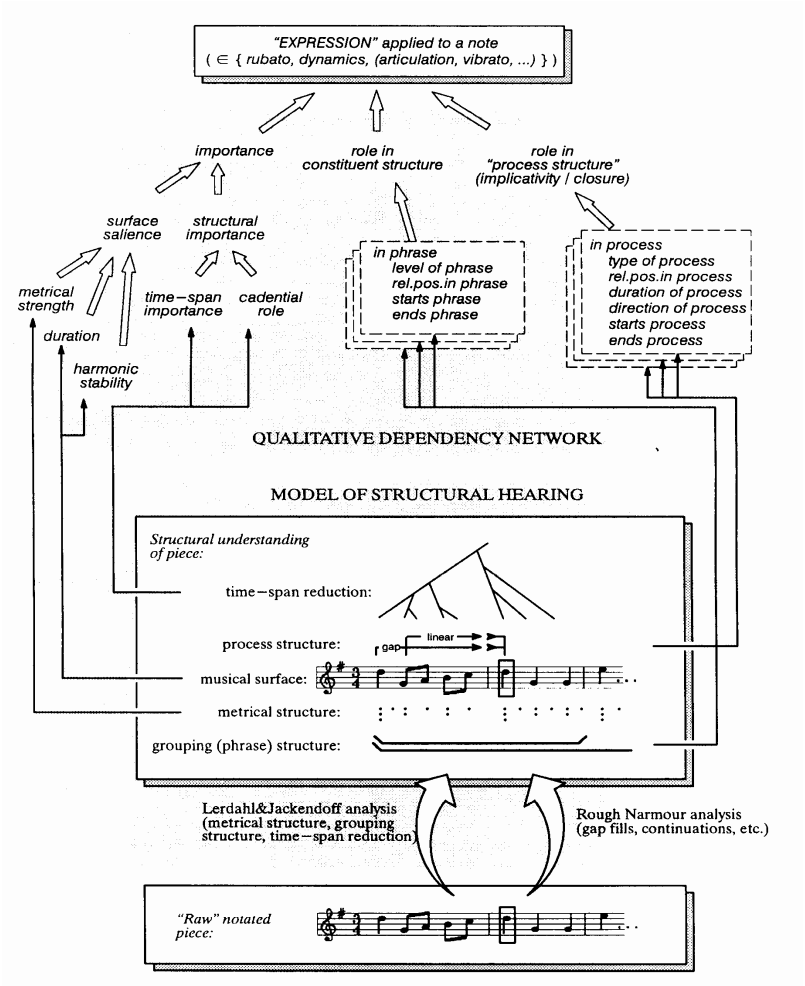


Fig. 2.9. Structural hearing (Widmer 1995)

The main dimensions in their model are: phrase structure, metrical structure, time-span reduction, and prolongational reduction (Widmer 1995). As shown in Fig. 2.9, the organization of musical structures is hierarchical. In contrast, Narmour concludes that linear connections between musical structures do not reflect the real nature of music. Widmer affirms Narmour's point of view, but he used a more simplistic approach in his experiment (Widmer 1995).

2.2.4 Rhythm Perception

It may be said that musicologists list a few elements of a musical piece. For example, Byrd and Crawford (2001) claim that the most informative are melody and rhythm, assigning about 50% of informativeness to a melody, 40% to rhythm and remaining 10% to the rest of such elements as harmony, dynamics, agogics, articulation, etc. (Byrd and Crawford 2002). Therefore, rhythm may be treated as one of the most fundamental components of music. The appearance of rhythm seems to be the first step in the evolution of a musical culture. In the days of ancient Greece and Rome, rhythm – tempo, measures and note duration – were defined by the kind of rhythmical recitation. Rhythmic, dynamic and harmonic notations differed greatly from modern musical notation. During the Renaissance, the tempo fixed at the beginning of a musical score was constant for the whole piece and denoted on the basis of '*taktus*' (*Latin*), the basic time signature, also referred to as '*integer valor notarum*' (*Latin*). Starting from that time, awareness of rhythm "grew up". In the Baroque period of music, rhythmic features started to be important, and the Classical period began with a new interest in rhythm. The modern period is marked by the strengthening of rhythmic features, exemplified in the compositions of Bartok and Stravinsky (The New Grove 1980).

The specific sequence of sound stimuli and pauses can be perceived as certain rhythm. The rhythm can be perceived if the presentations of sound stimuli are distributed in time interval of critical duration. Too short as well as too long duration of stimuli presentation precludes perceiving the rhythm.

Jingu (Nagai 1996) proposed an inner procedural model of time perception (Fig. 2.10). It is quite natural to assume the existence of an equivalent for a quartz oscillator of clocks in a human brain. This device serves as an internal clock. The internal clock and its pulse counter are considered to be used directly to evaluate temporal information and generate subjective time. The pacemaker in Jingu's model is based on the reverberating circuit in a brain. The pacemaker of this rhythm perception model is a black-box, but the same cycle of pulses as Jingu's model (4ms) is postulated as a standard. The counter is a device to count pulses generated by the pacemaker. This counter, dependent on the mode, counts from 4ms to 12ms pulses. The chrono-store in the rhythm perception model is an equivalent of echoic-memory in auditory perception. It is a temporary storage space for pulses from its pacemaker. The information stored is forwarded to the short-term memory (Nagai 1996).

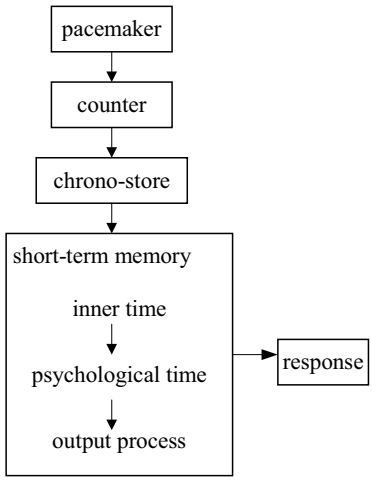


Fig. 2.10. Inner procedural model of time perception proposed by Jingu (Nagai 1996)

An interesting concept was proposed by Nagai (Nagai 1996). He observed that in the field of psychophysics, time shown by a physical clock is a physical continuum, while time felt by human beings is a psychological continuum. Both times change continuously in quantity, but the latter time needs consciousness, and is called subjective time. Time and rhythm perception shares its characteristics with language processing because both are independent of perceptual modes such as vision and speech.

The problem of developing a technique for finding rhythmic patterns is first of all a problem of definitions. Most definitions of rhythm, tempo and time are interdependent and are not explicitly explained. Special attention should be paid to the segmentation of rhythmic progressions with respect to timing accentuation (Tanguiane 1993). However, formulating rules for distinguishing accentuation is at the same time one of the most difficult problems. For this purpose, the notion of a rhythmic syllable – understood as a sequence of time events with the accent on the last event – was introduced (Tanguiane 1993). In this way, rhythmic syllables may be defined for a particular example in a musical piece. On the basis of this methodical approach, it is possible to elaborate a kind of rhythmic grammar that may be useful in rhythm perception modeling.

Music analysis is also the basis of systems that allow the automatic composition of a musical piece in a given style. The system created by Cope uses *motifs*, called also *signatures*. It is based on patterns of pitch intervals or rhythmic ratios that occur in more than one example of a style (Westhead and Smaill 1993). In the literature, a study on style recognition

may also be found. A system made by Westhead and Smaill reads data from standard MIDI code files and collects motifs into a style dictionary (Westhead and Smaill 1993). The data are split into two training sets, positive examples (musical pieces of the same style) and negative examples (pieces of different style). The style dictionary contains two-style subsets (Fig. 2.11) (Westhead and Smaill 1993). The two subsets have some overlap because there are motifs present in both of them. When a new test piece (*M*) is presented to this system, it attempts to classify the piece as being most similar - in terms of a motif - to either positive or negative examples in the dictionary. The similarity estimate is made relative to the dictionary mean. Since the dictionary has a different number of motifs of differing frequency in each style subset, it displays a bias towards one of the training sets. Other calculations which are based on Shannon's information theory are also made. Entropy estimates are made both with and without the use of motifs. The reduction in entropy that results from using motifs in the representation is then measured in order to suggest how well the style has been characterized. The set $\rho(M)$ represents motifs already existing and extracted from the dictionary when a new melody M is presented to the system. The frequencies with which these motifs occur in each style subset can be used to classify *M* (Westhead and Smaill 1993).

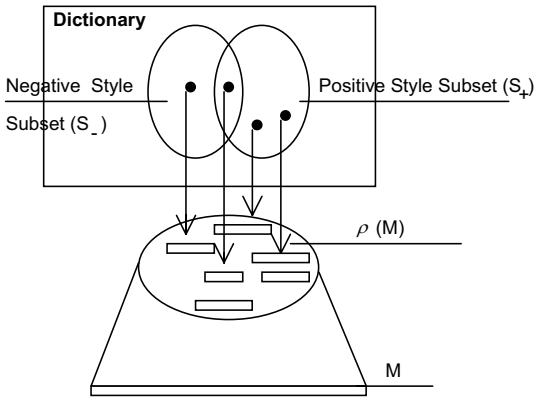


Fig. 2.11. Style recognition process (Westhead and Smaill 1993)

A style *S* is defined as a set of melodies M_i. A melody is a string of notes, each specified by pitch and duration. However, no account is taken of harmonies. The definitions of: $\rho(M_i)$, representing the set of all motifs present in the dictionary and in M, and $\mu(S)$, representing the set of all

motifs present in more than one melody in *S* are as follows (Westhead and Smaill 1993):

$$\mu(S) = \{w : \exists M_{i,} M_j \in S; i \neq j; w \in C(M_i) \wedge w \in C(M_j)\} \tag{2.18}$$

where $C(M_i)$ is the set of all possible motifs of M_i,

$$\rho(\mathrm{M}) = \{\mathrm{w} : \exists \mu(\mathrm{S_i}) \in D, w \in C(M) \wedge w \in \mu(S_i)\} \tag{2.19}$$

where *D* is a dictionary.

The entropy of a concept (next event in a piece) *H*(*X*) is defined as:

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x) \tag{2.20}$$

where the concept is a set with a probability function represented by a random variable *X*, such that $p(x) = P(X = x)$. The entropy represents the minimal expected number of bits required to specify an element of *X*. Hence, minimizing description length is achieved by minimizing the entropy (Westhead and Smaill 1993).

As a consequence of these assumptions, both dictionary data and melody data are extracted during the classification of musical pieces phase. As already mentioned, the data represented in the dictionary are as follows: the mean probability of all motifs in the dictionary that a motif is drawn from – the positive style subset; the variance of these probabilities; the total number of positive style motifs; the total number of negative style motifs. The melody data, on the other hand are represented by: the length (the number of pitch intervals in the melody); the mean probability of all motifs in the melody data that a motif comes from – the positive style subset; the variance of these probabilities; the number of motifs in the melody that only match motifs in the positive style subset; the number of negative motifs that matched; the significance (the probability that the melody mean value was arrived at by chance). The results obtained by Westhead and Smaill show that comparisons of style which are based on examples taken from different composers are more successful than when based only on the form specification (such as fugues, chorales, preludes, etc.), especially since the system has no representation of rhythms nor of the structure of the musical piece.

As shown in the musicological review given above, a musical fragment can be described by its form, rhythm, melodic contours, harmony, etc. These descriptors may then be used as attributes to be placed in a case-based musical memory, with values extracted from the chosen musical material. The system can detect similarities and discrepancies between musical events in order to provide a means of retrieving them. Automatic rec-

ognition of a musical style becomes one of the major topics within Music Information Retrieval (MIR, http://ismir2004.ismir.net/), thus it will also be reviewed in this context in the following sections.

References

McAdams S, Winsberg S (1999) Multidimensional scaling of musical timbre constrained by physical parameters. J Acoust Soc Am 105: 1273

Allott R (2004), Language and Evolution: Homepage, URL: http://www.percepp.demon.co.uk/evltcult.htm

Beerends J, Stemerdink J (1992) A Perceptual Audio Quality Measure Based on a Psychoacoustic Sound Representation. J Audio Eng Soc 40: 963-978

De Bruijn A (1978) Timbre-Classification of Complex Tones. Acustica 40: 108-114

Byrd D, Crawford T (2002) Problems of music information retrieval in the real world. Information Processing and Management 38: 249-272

Canazza S, de Poli G, Drioli C, Roda A, Vidolin A (2004) Modeling and control of expressiveness in music performance. Proc of the IEEE 92: 686-701

Cohen MA, Grossberg S, Wyse L (1995) A spectral network model of pitch perception. J Acoust Soc Am 98: 862-879

Cook PR (1999) Music, Cognition, and Computerized Sound, An Introduction to Psychoacoustics. MIT Press, Cambridge, Massachusetts, London, England.

Cosi P, De Poli G, Parnadoni P (1994) Timbre characterization with Mel-Cepstrum and Neural Nets. In: Proc of the ICMC'94, pp 42-45

Critchley M, Henson RA (eds) (1997) Music and the Brain. Heinemann, London

Durrant JD, Lovrinic JH (1977) Bases of Hearing Science. The Williams & Wilkins Company, Baltimore

Eitan Z, Granot RY (2004) Musical parameters and images of motion. In: Parncutt R, Kessler A, Zimmer F (eds). Proc of the Conference on Interdisciplinary Musicology, CIM04, Graz

Gelfand SA (1998) Hearing. An Introduction to Psychological and Physiological Acoustics. Marcel Dekker Inc, New York

Goldstein JL (1973) An optimum processor theory for the central formation of the pitch of complex tones. J Acoust Soc Am 54: 1496-1516

Grey JM (1977) Multidimensional perceptual scaling of musical timbres. J Acoust Soc Am 61: 1270-1277

Hatten R (1997-2002) Musical Gesture: on-line Lectures. Cyber Semiotic Institute, University of Toronto, URL: http://www.chase.utoronto.ca/epc/srb/cyber/hatout.html

Herrera P, Amatriain X, Battle E, Serra X (2000) Towards Instrument Segmentation for Music Content Description: A Critical Review of instrument classification techniques. In: Proc Int Symp on Music Information Retrieval, ISMIR'2000, Plymouth

Iverson P, Krumhansl CL (1993) Isolating the dynamic attributes of musical timbre. J Acoust Soc Am 94

Jensen K (2001) The timbre model, Workshop on current research directions in computer music, Barcelona

Jesteadt W, Bacon S, Lehman J (1982) Forward Masking as a Function of Frequency, Masker Level, and Signal Delay. J Acoust Soc Am 71: 950-962

Johannsen G (2004) Auditory displays in human-machine interfaces. Proc of the IEEE 92: 742-758

Kapust R (1989) Ein Gehörbezogenes Meßverfahren Zur Beurteilung Der Qualität Codierter Musiksignale. U.R.S.I. - Kleinheuerbacher Berichte, Band 33, Kleinheuerbach, pp 633 - 642

Kapust R (1992) A Human Ear Related Objective Measurement Technique Yields Audible Error and Error Margin. In: Proc of the 11th Intl Audio Engineering Society Conference. Portland, pp 191-202

Kostek B (1999) Soft Computing in Acoustics, Applications of Neural Networks, Fuzzy Logic and Rough Sets to Musical Acoustics, Studies in Fuzziness and Soft Computing. Physica Verlag, Heidelberg New York

Kostek B (2003) "Computing with words" Concept Applied to Musical Information Retrieval. Electronic Notes in Theoretical Computer Science 82, No 4

Kostek B (2004), Application of soft computing to automatic music information retrieval. J American Society for Information Science and Technology 55, No 12: 1108-1116

Kostek B, Czyzewski A (2001) Representing Musical Instrument Sounds for Their Automatic Classification. In: J Audio Eng Soc 49: 768-785

Krimphoff J, McAdams S, and Winsberg S (1994) Caracterisation du timbre des sons complexes. II Analyses acoustiques et quantification psycho-physique. J Phys 4: 625-628

Kurth E (1991) Selected writings. Rothfarb LA (trans & ed). Cambridge University Press, Cambridge

Larson S (1997) Musical Forces and Melodic Patterns, Theory and Practice 22-23, pp 55-71

McLaughlin T (1970) Music and Communication. Faber and Faber, London

Leach J, Fitch J (1995) Nature, Music, and Algorithmic Composition. J Computer Music 19

Lederman SJ, Klatzky RL (1996) Haptic aspects of motor control. in: Boller & Grafman (eds) Handbook of Neuropsychology 11: Jeannerod M (ed) Action and cognition, Elsevier Science Publishers, Amsterdam

Lederman SJ (2004), Introduction to Haptic Display: Psychophysics. Queen's University, Kingston, URL: http://haptic.mech.northwestern.edu/intro/psychophysics/lims.html

Lidov D (1999) Elements of Semiotics. St Martin's Press, New York

Meddis R, Hewitt MJ (1991) Modeling the identification of concurrent vowels with different fundamental freuencies. J Acoust Soc Am 54: 610-619

Meyer J (1993) The Sound of the Orchestra. J Audio Eng Soc 41: 203-212

Moore BCJ (1989) An Introduction To The Psychology Of Hearing. Academic Press, New York

Moore BCJ (1996) Masking in the Human Auditory System. In: Gilchrist N, Grewin C (eds) Collected Papers on Digital Audio Bit-Rate Reduction. Audio Engineering Society, pp 9-19

Moorer BCJ (1997) An Introduction to the Psychology of Hearing, 4th edn. Academic Press

Nagai K (1996) A study of a rhythm perception model. URL: http://www.tsuyama-ct.ac.jp/kats/papers/kn8/kn8.htm

Parsons D (1975) The Directory of Tunes and Musical Themes, Spencer Brown, Cambridge

Patterson RD, Holdsworth J (1996) A functional model of neural activity patterns and auditory images. Advances in Speech, Hearing and Language Processing

Penner M (1980) The Coding of Intensity and the Interaction of Forward and Backward Masking. J Acoust Soc Am 67: 608-616

Penner M, Shiffrin R (1980) Nonlinearities in the Coding of Intensity Within the Context of a Temporal Summation Model. J Acoust Soc Am 67: 617-627

Pohlmann KC (1995) Principles of Digital Audio. McGraw-Hill Inc, New York

De Poli G, Piccialli A, Roads C (eds) (1991) Representations of Musical Signals. MIT Press, Cambridge

De Poli G, Canazza S, Drioli C, Roda A, Vidolin A, Zanon P (2001) Analysis and modeling of expressive intentions in music performance. In: Proc of the Intern Workshop on Human Supervision and Control in Engineering and Music. Kassel, URL: http://www.engineeringandmusic.de/workshop

Pollard HF, Jansson EV (1982) A Tristimulus Method for the Specification of Musical Timbre. Acustica 51: 162-171

Pratt RL, Doak PE (1976) A subjective rating scale for timbre. J Sound and Vibration 45: 317-328

Pratt RL, Bowsher JM (1978) The subjective assessment of trombone quality. J Sound and Vibration 57: 425-435

Reuter C (1996) Karl Erich Schumann's principles of timbre as a helpful tool in stream segregation research. In: Proc II Intern Conf on Cognitive Musicology, Belgium

Schroeder M, Atal BS, Hall JL (1979) Objective Measure Of Certain Speech Signal Degradations Based On Masking Properties Of Human Auditory Perception. In: Lindblom, Öhman (eds) Frontiers of Speech Communication Research. Academic Press, New York, pp 217-229

Scruton R (1997) The Aesthetic of Music. Clarendon Press, Oxford

Suzuki K, Hashimoto S (2004) Robotic interface for embodied interaction via dance and musical performance. Proc of the IEEE 92: 656

Tanguiane AS (1993) Artificial Perception and Music Recognition. Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin

Terhardt E (1979) Calculating Virtual Pitch. J Hearing Research 1: 155-182

Terhardt E (1992) The Spinc Function For Scaling Of Frequency In Auditory Models. Acustica 77: 40-42

The New Grove Dictionary of Music and Musicians (1980). Macmillan Publishers, Sadie S (ed), London Washington Hong Kong

Thiede T (1999) Perceptual Audio Quality Assessment using a Non-Linear Filter Bank. Ph.D. thesis, Technical University of Berlin

Thiede T, Treurniet W, Bitto R, Schmidmer Y, Sporer T, Beerends J, Colomes C, Keyhl M, Stoll G, Brandenburg K, Feiten B (2000) PEAQ - The ITU Standard for Objective Measurement of Perceived Audio Quality. J Audio Eng Soc 48

Tsoukalas D, Mourjopoulos J, Kokkinakis G (1997) Perceptual Filters for Audio Signal Enhancement. J Audio Eng Soc 45: 22-36

Vercoe B, Gardner W, Schreier E (1998) Structured Audio: Creation, Transmission, and Rendering of Parametric Sound Representations. Proc IEEE 86: 922-940

Walmsley PJ (2000) Signal Separation of Musical Instruments - Simulation-based methods for musical signal decomposition and transcription. Ph.D. thesis, University of Cambridge

Wessel DL (1979) Timbre Space as a Musical Control Structure. J Computer Music 3: 45-52

Westhead MD, Smaill A (1993) Automatic Characterisation of Musical Style, in Music Education: An Artificial Intelligence Approach. Smith M, Smaill A, Wiggins GA (eds) Workshops in Computing. Springer-Verlag, Edinburgh, pp 157-170

Widmer G (1995) Modeling the Rational Basis of Musical Expression. J Computer Music 19: 76-96

Wightman FL (1982) The pattern-transformation model of pitch. J Acoust Soc Am 71: 679-688

Zölzer U (1996) Digitale Audiosignalverarbeitung. BG Teubner, Stuttgart

Zwicker E (1961) Subdivision of the audible frequency range into critical bands. J Acoust Soc Am

Zwicker E, Fastl H (1990) Psychoacoustics, Facts and Models. Springer Verlag, Berlin Heidelberg

Zwicker E, Feldkeller R (1967) Das Ohr als Nachrichtenempfänger. Hirzel Verlag, Stuttgart

Zwicker E, Terhardt E (1980) Analytical Expressions For Critical Bandwidth As A Function Of Frequency. J Acoust Soc Am 68: 1523-1525

Zwicker E, Zwicker T (1991) Audio Engineering and Psychoacoustics: Matching Signals to the Final Receiver, the Human Auditory System. J Audio Eng Soc 39: 115-126

URL: http://ismir2004.ismir.net/ Music Information Retrieval website (2004)

3 INTELLIGENT MUSICAL INSTRUMENT SOUND CLASSIFICATION

3.1 Problem Overview

This chapter is devoted to intelligent classification of the sound of musical instruments. Although it is possible, and in some applications sufficient to process musical data based on statistical methods, clearly such an approach does not provide either computational or cognitive insight. The principal constituents of intelligent computation techniques are data mining, machine learning, knowledge discovery algorithms, decision-systems, learning algorithms, soft computing techniques, artificial intelligence – some of these notions have become independent areas, and some of them are nearly synonymous. Data mining, also referred to as Knowledge Discovery in Databases – KDD, has been defined by Frawley et al. as "the nontrivial extraction of implicit, previously unknown, and potentially useful information from data". Soft computing aims at using machine learning to discover and to present knowledge in a form, which is easily comprehensible to humans. Physical systems described by multiple variables and parameter models having non-linear coupling, frequently occur in the fields of physics, engineering, technical applications, economy, etc. The conventional approaches for understanding and predicting the behavior of such systems based on analytical techniques can prove to be very difficult, even at initial stages of establishing an appropriate mathematical model. The computational environment used in such an analytical approach is perhaps too categorical and inflexible to cope with the complexity of physical systems of the real world. It turns out that when dealing with such systems, one has to face a high degree of uncertainty and to tolerate imprecision. Trying to increase precision can be very costly.

Lotfi A. Zadeh separates hard computing based on binary logic, crisp systems, numerical analysis and crisp software, from soft computing based on fuzzy logic, neural nets and probabilistic reasoning (http://www.soft-computing.de/def.html). The former is characterized by precision and the

Bożena Kostek: *Perception-Based Data Processing in Acoustics*, Studies in Computational Intelligence (SCI) **3**, 39–186 (2005)
www.springerlink.com

latter, by – approximation. Although in hard computing, imprecision and uncertainty are undesirable, in soft computing the tolerance for imprecision and uncertainty is exploited to achieve tractability, lower cost, high *Machine Intelligence Quotient* (MIQ) and economy of communication.

There are several definitions concerning soft computing as a domain of science. The most widely known and most often used methods of soft computing (or computational intelligence) are neural networks, multivalued logic, fuzzy sets and fuzzy logic, Dempster-Shafer theory, rough sets, probabilistic reasoning, evolutionary computation, etc. Particular attention was paid in this work to neural networks, fuzzy logic and rough sets. Neural networks may be treated as tools for modeling dependencies between variables. Fuzzy and rough sets are formal methods for dealing with uncertainty. These techniques are reviewed further in this chapter, because when applied to classification tasks they provide a kernel to decision algorithms. A particular justification for the application of decision systems in this area is provided by the fact that the management of uncertainty in acoustics should be based on the knowledge of experts – the best criterion for assessing the acoustical quality of music.

This chapter does not provide a comprehensive review of the vast research in these areas. The primary purpose was to list some techniques that are applicable to the field of Music Information Retrieval (Downie 2003; http://ismir2004.ismir.net/). In addition, this Chapter is concerned with processing techniques applied to musical signal processing and feature extraction, therefore it will begin with a review of a number of principles so that the following sections can proceed without the need for basic definitions and concepts.

Finally, several other factors should be considered when selecting a technique for an application to a specific problem: efficiency, complexity, memory size, the ability to generalize, etc. Therefore, in some applications a hybrid approach is chosen and refined to overcome the limitations of one technique by combining it with another more effective in specific tasks.

3.2 MUSICAL SIGNAL PROCESSING

3.2.1 Spectral Analysis

Apart from most frequently used FFT transform, there are some other transforms that allow analysis in the frequency domain, such as Walsh-

Hadamard transform, which involves analysis in terms of square waves of different frequencies, cosine transform, (modified cosine transform), McAulay & Quatieri algorithm (McAulay and Quatieri 1986), etc. Furthermore, there exist spectral estimation methods, among others classical ones based on parametric methods. These methods refer to a variety of equivalent formulations of the problem of modeling the signal waveform, the differences underlying these formulations concern mostly the details of computations. In the literature methods based on autocorrelation, covariance, maximum entropy formulation are often cited. Algorithms known as Prony, Yale-Walker, Burg, Durbin, Pisarenko (Kay 1988; Marple 1987), etc., provide practical spectral signal estimation. The above cited methods are based on linear processes. They are efficient enough also when extending to the identification of adaptive dynamic models. This is because with suitable pre-processing, a non-linear problem may often be converted into a linear one. However, as the processes become more complex, a sufficiently correct non-linear input-output behavior is more difficult to obtain using linear methods. Lately, in the literature on control system identification methods based on input-output models for non-linear systems, both deterministic and stochastic appeared. They are known as NARMAX (**N**on-linear **A**uto**R**egressive **M**oving **A**verage with E**X**ogenenous input) and NARX (**N**on-linear **A**uto**R**egressive with E**X**ogenenous input) models (Billings and Chen 1989; Leonarties and Billings 1985).

Below, some spectrum estimation methods will be reviewed in order to show that in some cases such methods are more accurate than FFT analysis. Generally, a large class of parametric methods fall under the category of spectral estimation. Therefore, some chosen methods of spectral estimation that are based on power series models are reviewed in this study, namely Autoregressive (AR), Moving Average (MA), and Autoregressive – Moving Average (ARMA) models. These methods are often described in terms of Zeros-Poles approximations, i.e. the MA model belongs to 'all-zero' methods, while AR belongs to 'all-poles'. Some examples of analyses using AR, MA, and ARMA processes will be given in order to show that these methods may be useful for the analysis of spectra of musical sounds (Kay 1988; Marple 1987).

Spectral estimation is a three-fold method. First, the appropriate model is chosen. Then, model parameters are computed. Finally, in the third phase, computed model parameters provide coefficients for the evaluation of the PSD (Power Spectral Density) function.

Let $u[n]$ be the input and $x[n]$ be the output signals. These signal sequences are related by following expression:

$$x[n] = -\sum_{k=1}^{p} a[k]x[n-k] + \sum_{k=0}^{q} b[k]u[n-k] \tag{3.1}$$

where: a, b are model parameters and a pair (p,q) represents the order of the model. Eq. (3.1) is known as the ARMA model.

The transmittance $H(z)$ between $u[n]$ and $x[n]$ for the ARMA process is defined as:

$$H(z) = \frac{B(z)}{A(z)} \tag{3.2}$$

where: $A(z)$ - the z-transform of the AR part of the process, $A(z) = \sum_{k=0}^{p} a[k]z^{-k}$, $B(z)$ - the z-transform of the MA part of the process, $B(z) = \sum_{k=0}^{q} b[k]z^{-k}$ and $a[k]$, $b[k]$ are the coefficients of the autoregression function and the moving average, respectively. It is assumed that $A(z)$ can have poles that lie inside the unit z-circle in order to guarantee the stability of the system.

It is known that z-transform of the autocorrelation function $P_{xx}(z)$ is equal to the power spectrum $P_{xx}(f)$, on the condition that $z = \exp(j2\pi f)$ for $-\frac{1}{2} \le f \le \frac{1}{2}$.

If all coefficients $a[k]$ except $a[0] = 1$ equal zero in the ARMA process, then:

$$x[n] = \sum_{k=0}^{q} b[k]u[n-k] \tag{3.3}$$

and this process is known as the MA of the order q, and its power spectrum is given as (Marple 1987):

$$P_{MA}(f) = \sigma^2 \left| B(f) \right|^2 \tag{3.4}$$

on the condition that $u[n]$ is a white noise sequence with mean value equal 0 and variance σ^2 is equal to the white noise power density.

On the other hand, if all coefficients $b[k]$ except $b[0] = 1$ equal zero in the ARMA process, then:

$$x[n] = \sum_{k=1}^{p} a[k]x[n-k] + u[n] \tag{3.5}$$

and this process is known as the AR (autoregression model) of the order p, and its power spectrum is:

$$P_{AR}(f) = \frac{\sigma^2}{\left|A(f)\right|^2} \tag{3.6}$$

Based on the Wiener-Khinchin theorem which says that the Fourier transform of the autocorrelation is equal to the power spectrum, it is possible to express the power spectrum of the MA process as follows:

$$P_{MA}(f) = \sum_{k=-q} r_{xx}[k]\exp(-j2\pi fk) \tag{3.7}$$

The same analogy can be applied to the AR and ARMA processes.

It is known that under the condition that the power spectrum is infinite, then for the ARMA(p,q) process the AR(p) and MA(q) equivalent models do exist.

Provided $h[k] = 0$ for $k < 0$, then the autocorrelation function $r_{xx}[k]$ for the ARMA process is as follows:

$$r_{xx}[k] = \begin{cases} -\sum_{l=1}^{p} a[l]r_{xx}[k-l] + \sigma^2 \sum_{l=0}^{q-k} h^*[l]b[l+k] & for \quad k = 0,1,...,q \\ -\sum_{l=1}^{p} a[l]r_{xx}[k-l] & for \quad k \geq q+1 \end{cases} \tag{3.8}$$

where: $h[l]$ is actually the impulse response of the system with transfer function $H(z)$.

Providing that $b[l] = \delta[l]$, then the autocorrelation function for the AR process is:

$$r_{xx}[k] = -\sum_{l=1}^{p} a[l]r_{xx}[k-l] + \sigma^2 h^*[-k] \tag{3.9}$$

Additionally, if $h^*[-k] = 0$ for $k > 0$ and $h^*[0] = [\lim_{z\to\infty} H(z)]^* = 1$, then:

$$r_{xx}[k] = \begin{cases} -\sum_{l=1}^{p} a[l] r_{xx}[k-l] & for \quad k \geq 1 \\ -\sum_{l=1}^{p} a[l] r_{xx}[-l] + \sigma^2 & for \quad k = 0 \end{cases} \tag{3.10}$$

The above equations are known as Yule-Walker's equations. For computational purposes, the above equations are often given in matrix form:

$$\underbrace{\begin{bmatrix} r_{xx}[0] & r_{xx}[-1] & \cdots & r_{xx}[-(p-1)] \\ r_{xx}[1] & r_{xx}[0] & \cdots & r_{xx}[-(p-2)] \\ \vdots & \vdots & \ddots & \vdots \\ r_{xx}[p-1] & r_{xx}[p-2] & \cdots & r_{xx}[0] \end{bmatrix}}_{\mathbf{R}_{xx}} \begin{bmatrix} a[1] \\ a[2] \\ \vdots \\ a[p] \end{bmatrix} = -\begin{bmatrix} r_{xx}[1] \\ r_{xx}[2] \\ \vdots \\ r_{xx}[p] \end{bmatrix} \tag{3.11}$$

Correspondingly, for the MA process, when $a[l] = \delta[l]$ and $h[l] = b[l]$, the autocorrelation function is as follows:

$$r_{xx}[k] = \begin{cases} \sigma^2 \sum_{l=0}^{q-k} b^*[l] b[l+k] & for \quad k = 0,1,...,q \\ 0 & for \quad k \geq q+1 \end{cases} \tag{3.12}$$

It is known from literature that the AR and equivalent ARMA models provide an accurate representation for underlying power spectra which have sharp spectral features (Kay 1988). Therefore, most of the carried out musical sound analyses aimed at testing algorithms that are based on AR and ARMA processes. In order to estimate the power spectral density in the AR model, estimation methods other than the autocorrelation method are also used, namely: covariance, modified covariance, Burg's method, RMLE (Recursive Maximum Likelihood Estimation) method, etc. It should be remembered that both the AR and MA processes may be treated as specific cases of the ARMA process. Starting from the ARMA process, it is possible to estimate the power spectra of these processes by assuming that the order of the MA model, denoted as q, is equal to 0 in the AR process, while the order of the AR model, expressed as p, equals 0 in the MA process (Kay 1988; Marple 1987; Press et al 1986).

It should be noted that the choice of the frame length (N) and the determination of the model order are a very important consideration in implementation. Clearly, for the autocorrelation method N should be on the order of several pitch periods to ensure reliable results. To effectively evaluate the model order, it is necessary to use one of commonly used techniques and criteria. Basically, the model order is assumed on the basis of the computed prediction error power. First of all, a minimum of the so-called *Final Prediction Error* (*FPE*) function defined as follows:

$$FPE(k) = \frac{N+k}{N-k}\hat{\rho}_k \tag{3.13}$$

where $\hat{\rho}_k$ is the variance of the white noise process (prediction error power) serves as such a criterion. Another criterion, known in the literature as *Akaike Information Criterion* (*AIC*) is expressed below (Kay 1988):

$$AIC(k) = N \ln \hat{\rho}_k + 2k \tag{3.14}$$

is often used. The chosen model order is the computed minimum of the expression given above. The expression $2k$ is often substituted by the factor $k \cdot \ln N$ due to the fact that computed order is too high, providing a big value of N. One may use also *Bayesian Information Criterion* (*BIC*) as a measure of the goodness-of-fit of the model.

Theoretically, ARMA models being zero-pole signal representation are more accurate than AR or MA models, however in practical musical sound analysis it might be proven that pole-representation (AR models) may be more useful, because it represents the resonant structure of a musical instrument body. On the basis of performed analyses, it may be said that spectral approximation obtained on the basis of the AR model, regardless of the method of analysis, is more accurate than when ARMA or MA models were used. The number of samples used in the analyses (N) influences the quality of the spectral analysis. At the same time, a decrease in the number of samples from 512 to 256 often results in better resolution. The crucial point of analysis is, however, the assignment of the order of the model; assumed value of the model order in all parametric methods is of high importance. A more general conclusion concerns the number of sound samples (N); namely, for $N<512$, it is more convenient to use parametric methods because they are more accurate than the FFT analysis, while for $N>512$, the FFT analysis gives better spectrum estimation.

Below, some examples of analyses obtained on the basis of some parametric methods that were implemented algorithmically in the Multimedia Systems Department are shown. In Fig. 3.1, three spectral analyses of flute, violin and piano sounds (C6) are shown for sounds belonging to dif-

ferent instrument groups. What is especially important, these sounds contain differentiated amount of noise depending on the sound generating mechanism. The analyses shown below were performed using the autocorrelation method.

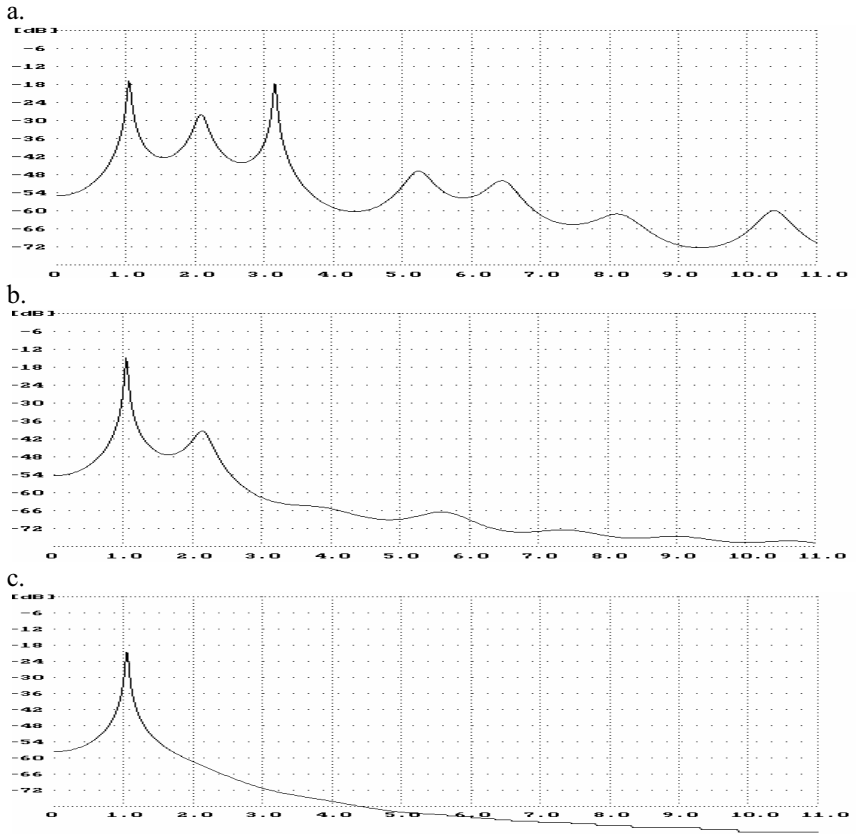


Fig. 3.1. Spectrum of C6 violin (a), piano (b), and flute (c) sounds, autocorrelation method $p = 28$ (AR model), $N = 128$

As seen from analyses the method presented accurately estimates the pitch of the analyzed sounds. Although the first harmonic is clearly in evidence in all plots, it can be seen that the autocorrelation method reveals fewer peaks than it is expected; higher spectrum partials of less energy are often disregarded and are not shown in the analysis. The easiest for spectrum estimation with regard to these three instruments is a violin sound, since its harmonics are of high energy. On the other hand, only two harmonics are identified for piano, other harmonics are below the noise background. The similar situation happens for the flute sound, in such a case only the first harmonic is visible in spectral analysis, other harmonics, hav-

ing not sufficient energy, are hidden by noise. The value of 28 assigned to *p* parameter causes the algorithm to treat harmonics of less energy as noise partials.

In order to compare spectra that were obtained using parametric methods with the corresponding one obtained on the basis of the FFT transform, examples of such analyses of three musical sounds are shown in Fig. 3.2-3.7. A direct comparison of the spectra using FFT transform with these obtained on the basis of the AR model (modified covariance method) is shown for other musical sounds. As seen, the parametric representation leads to a very good estimation of subsequent sound harmonics, however the signal to noise ratio (SNR) is an important factor influencing the quality of the spectral estimation. When comparing two violin sounds C6 B5 (Fig. 3.2) and B5 (Fig. 3.4) obtained in parametric-based spectral analysis it may be concluded, that both the model order (*p*,*q*) and number of sound samples (*N*) should be carefully chosen in the analysis. For the example in Fig. 3.2 most sound harmonics are better resolved than in the case of the B5 sound analysis. Interesting case is the analysis of the pipe organ sound. For sounds that contain natural noise the modified covariance method does not get a problem with estimating higher harmonics, as was the case with the autocorrelation method (Kostek 1999).

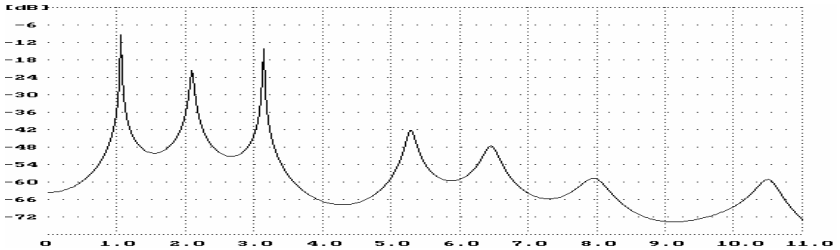


Fig. 3.2. Spectrum of C6 violin sound, modified covariance method p = 28, N= 512

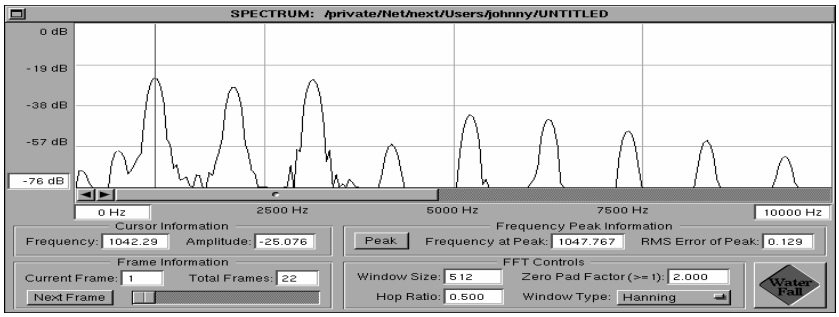


Fig. 3.3. FFT analysis of a C6 violin sound, Hanning window

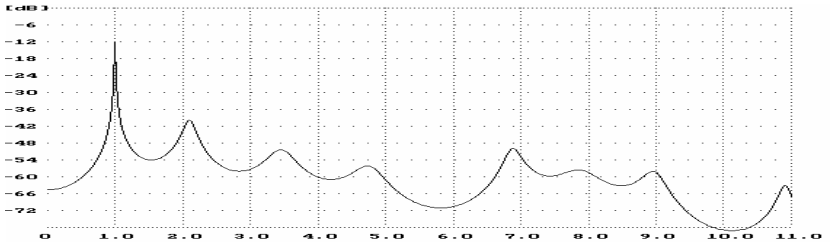


Fig. 3.4. Spectrum of B5 violin sound, modified covariance method p = 28, N= 512

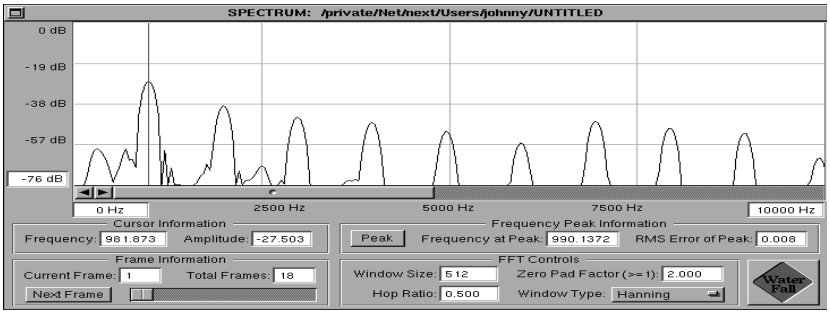


Fig. 3.5. FFT analysis of a B5 violin sound, Hanning window

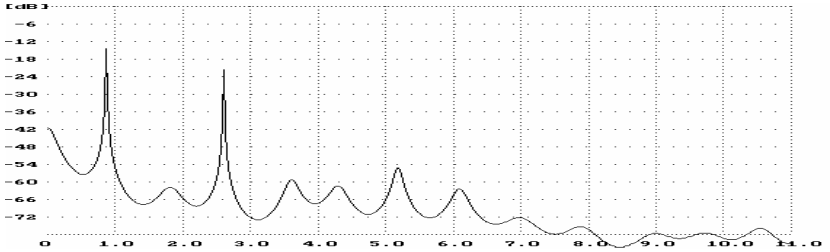


Fig. 3.6. Spectrum of A5 pipe organ sound, modified covariance method (AR model), p = 28, N = 512

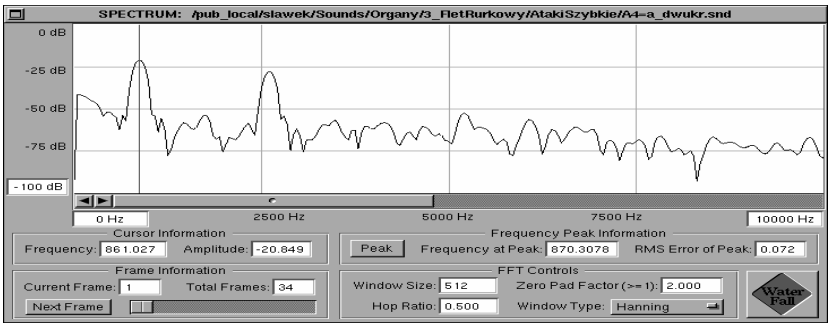


Fig. 3.7. FFT analysis of a A5 pipe organ sound, Hanning window

The overall observation is such as follows:

- presented methods accurately estimate the pitch of the analyzed sound;
- although the first harmonic is clearly in evidence in all plots, it can be seen that both ARMA and MA model-based methods reveal fewer peaks than in the adequate FFT analysis;
- parametric methods based on the AR model give very sharp and high maxima for the spectrum partials of high energy as compared with the adequate FFT analysis, especially in the case of the low partials, but contrarily, higher spectrum partials of less energy are often disregarded and are not shown in the analysis.

3.2.2 Wavelet Analysis

One of most popular methods in the domain of signal processing is time-frequency signal analysis. This is due to the fact that signal processing becomes an important tool in domains such as seismology, medicine, speech, vibration acoustics, etc. (Chui et al 1994; Choi and Williams 1989; Evangelista 1993; Guillemain and Kronland-Martinet 1991; Mallat 1991; Meyer 1992; Wilson et al 1992). Most real signals that are analyzed in practice are of a non-stationary character, that is why their conventional approximation by means of stationary signals using classical frequency estimation methods is not faithful enough and may cause even gross errors.

One of the main advantages of wavelets is that they offer a simultaneous localization in time and frequency domain. This is also simply an alternative way of describing a signal in the frequency domain. Such a description in the frequency domain provides a more parsimonious representation than the usual one on a grid in the time domain.

Originally, the time-frequency analysis was proposed by Gabor. He showed that a signal apart from time and frequency representation can have a two-dimensional representation. He proposed a technique that leads to the frequency analysis by partitioning signal into fragments and applying a window function. The performed convolution process used a bell-shaped time envelope, generated by the Gaussian method (De Poli et al 1991):

$$p(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx \tag{3.15}$$

Gabor's time-frequency signal analysis method was reexamined by Grossmann and Morlet, and later by Kronland-Martinet and provide the

basis of the wavelet transform (Evangelista 1993; Guillemain and Kronland-Martinet 1991).

Wavelet transformation is a powerful tool for time-frequency signal analysis (Chui et al 1994; Genossar and Porat 1992). This transform is especially useful when it is necessary to characterize transient phenomena. By using adequately dilated or contracted copies of a mother function (see Fig. 3.8 and 3.9), it enables the analysis of various frequency bands of the given signal with different resolutions. This solves the problem of obtaining a high resolution simultaneously in the time- and frequency-domains (De Poli et al 1991; Orr 1993).

The elementary wavelet functions $g_{b,a}(t)$ that are subjected to a change of scale are copies of a wavelet mother function $g(t)$:

$$g_{b,a}(t) = \frac{1}{a} \cdot g\left(\frac{t-b}{a}\right) \tag{3.16}$$

where b is any real number, and a is the rescaling coefficient, $a>0$.

The frequency localization is given by the Fourier transform of the $g_{b,a}(t)$ function:

$$\hat{g}_{b,a}(\omega) = \hat{g}\left(a \cdot \omega\right) \cdot e^{j \cdot \omega \cdot b} \tag{3.17}$$

where $\hat{g}_{b,a}(\omega)$ is the Fourier transform of the function $g_{b,a}(\omega)$.

The localization depends on the parameter a. The resulting decomposition will consequently be at $\Delta\omega / \omega = \text{constant}$. For this reason, wavelets can be interpreted as impulse responses of constant Q-filters.

Assuming that a signal is composed of a set of elementary functions, the wavelet transform is thus given by:

$$S(b,a) = \left\langle g_{b,a} \middle| s \right\rangle = \int \overline{g}_{b,a}(t) \cdot s(t) dt = \frac{1}{\sqrt{a}} \int \overline{g}_{b,a}\left(\frac{t-b}{a}\right) \cdot s(t) dt \tag{3.18}$$

where the bar denotes complex conjugation.

The Fourier transform tends to decompose an arbitrary signal into harmonic components, whereas the wavelet transform allows free choice of elementary function (De Poli et al 1991). This feature of the wavelet transform seems of importance because it is possible to carry out musical sound analyses that are specific for a given instrument (i.e. mother function derived from the analysis of an instrument structure), therefore using differentiated mother functions.

In Fig. 3.10 time-frequency resolution of DWT (*Discrete Wavelet Transform*) and STFT (*Short-Time Fourier Transform*) analyses is shown. The DWT of signal $x[n]$ can be presented as:

$$DWT(a,n) = \frac{1}{\sqrt{a}} \sum_k h\left(\frac{k}{a} - n\right) \cdot x(k) \tag{3.19}$$

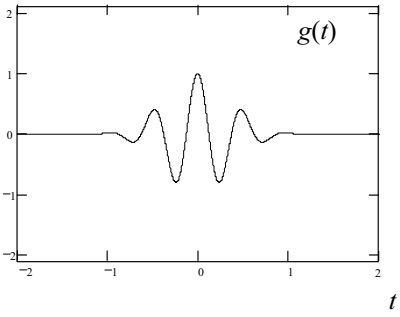


Fig. 3.8. Mother wavelet function

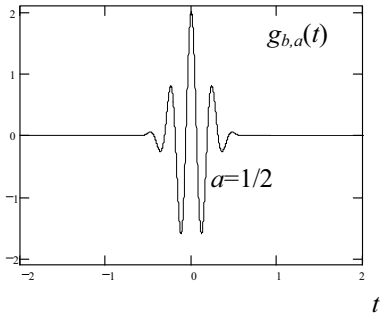


Fig. 3.9. Elementary wavelet scaled with $|a|<1$

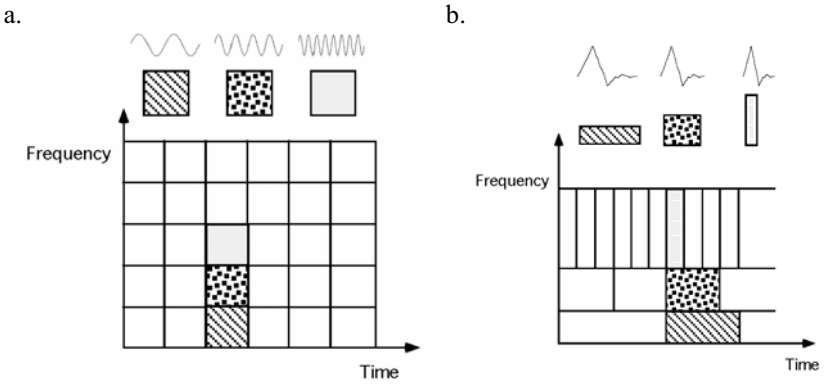


Fig. 3.10. Time-frequency resolution of STFT (a) and DWT (b) transforms

On the basis of the wavelet transform, it is possible to define certain parameters, such as distribution of energy or measure of discontinuities.

A wavelet transform may be implemented as a bank of filters that decompose a signal into multiple signal bands (see block diagram in Fig. 3.11). It separates and retains signal features in one or a few of these subbands. Thus, the main advantage of using the wavelet transform is that signal features can be easily extracted. In many cases, a wavelet transform

outperforms the conventional FFT transform when it comes to feature extraction.

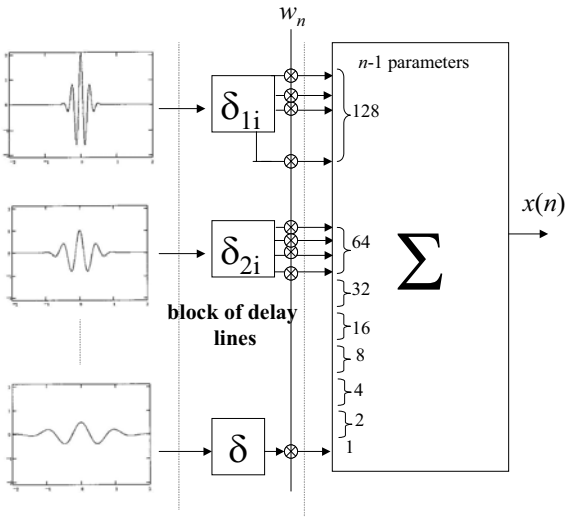


Fig. 3.11. Presentation of the wavelet transformation implementation

3.2.3 Pitch Detection Algorithms

Pitch detection is one of the most difficult tasks in speech and musical signals processing and has been studied in many publications for many years (Brown 1992; Beauchamp 1993a; Czyzewski et al 2002; Klapuri 1999; Maher and Beauchamp 1994; Proakis and Manolakis 1999). It is due to the fact that acoustic signals are non-stationary, i.e. pitch and amplitudes of their harmonics vary in time. In many cases significant noise disturbances are contaminating analyzed signals, making pitch estimation even more difficult. Owing to these reasons a universal solution for the problem does not seem to exist and the proposed pitch detection algorithms (PDAs) vary often in accordance with different requirements and applications. In the domain of processing acoustic signals, two major applications of pitch detection are known: pitch determination of speech signals and pitch determination of musical signals. In the case of speech signals (Rabiner et al 1976; McGogenal et al 1977; Hess 1983), it is very important to determine pitch almost instantaneously, which means that processed frames of the signal must be small. This is because voiced fragments of speech may be very short, with rapidly varying pitch. In the case of musical signals, voiced (pitched) fragments are relatively long and pitch fluctuations are

lower. This enables the use of larger segments of a signal in the pitch estimation procedure. But for both applications, efficient pitch detection algorithm should estimate pitch periods accurately and smoothly between successive frames, and they should produce pitch contour that has high resolution in the time-domain.

In general, estimating frequency of digital signals can be divided into two main categories. The first category focuses on estimating the sinusoidal, or harmonic time domain properties of signals in a noisy environment, however it is often assumed that some properties of a signal and noise are known (Rife and Boorstyn 1974, 1976). Practical applications for this group of algorithms are operating in stationary, or slowly changing conditions, or on non-audio signals. This group of algorithms deals with e.g. radar antenna signals, sonar signals, digital modem signals, a wide variety of telecommunication signals, etc.

Algorithms of the second category operate on audio signals and are responsible for detecting the pitch of sounds, i.e. they simulate human perception of sounds in terms of perceiving signal frequency.

As suggested by Rabiner et al. (1976) pitch detection algorithms can be roughly divided into three broad categories:

- PDAs utilizing time-domain properties of signals.
- PDAs utilizing frequency-domain properties of signals.
- PDAs utilizing both: time and frequency properties of signals.

Time-domain related algorithms (Wize et al 1976), such as AMDF (Average Magnitude Difference Function) (Quian and Kimaresan 1966; Talkin 1995; Ying et al 1996) and modified AMDF methods (Chang and Su 2002; Dziubinski and Kostek 2004; Kostek 2004a, 2004b; Medan et al 1988; Mei et al 2001; Szczerba and Czyzewski 2005; Zhang et al 2002) analyze waveform properties, and estimate pitch period by analyzing zero-crossing, or peaks and valleys of signals. Autocorrelation based algorithms (Rabiner and Schafer 1978) work similarly, operating however, on the autocorrelation signal. Such approach is based on the assumption that the processed signal is of the quasi-periodic type, and its consecutive periods are similar to each other. In many cases, appropriate preprocessing is applied in order to enhance periodicity properties of signals. Still, quite often *gross pitch errors* and relatively large *fine pitch errors* constitute a major problem for this group of PDAs, due to differences between consecutive periods. Suitable modifications have, however, been proposed, significantly improving the time-domain PDAs performance (Chang and Su 2002; Medan et al 1988, 1991; Mei et al 2001; Zhang et al 2002). In general, time-domain related algorithms are very useful in real time applications for pitch estimation of speech signals. They are computationally effi-

cient and can operate on small signal blocks, introducing short delay to calculated results.

Frequency-domain related algorithms operate on a signal spectrum (McAulay and Quatieri 1990; d'Alessandro 1995; Ahn and Holmes 1997), or on a nonlinearly processed representation of its cepstrum (Noll 1967), also on a so-called ACOLS – Autocorrelation Of Log Spectrum (Kunieda et al 1996) – with the assumption that the fundamental frequency of the signal and its harmonics are represented by some of the spectrum peaks. This group of algorithms usually requires larger blocks of the input signal to provide appropriate resolution of the analyzed spectrum (cepstrum). This is an important issue, especially in the case of low pitched signals, and it seems to be the major disadvantage of this group of PDAs. On the other hand, these algorithms are effective for signals containing only few higher order harmonics and significant noise level (radio-transmitted speech signals with strong low frequency noise content, noisy trumpet sounds, etc.), where time-domain related algorithms are more likely to fail. In addition, by analyzing relations between peaks representing harmonic frequencies, it is possible to effectively avoid octave errors (*gross errors*), as reported later in this paper.

The third group of PDAs incorporates properties of time and frequency-domain algorithms. In some cases AMDF or autocorrelation methods are applied, and some information is gathered from the calculated spectrum, in order to decrease the possibility of estimation errors (Hu 2000; Kasi and Zahorian 2002), resulting in a more accurate pitch tracking. Such operations usually require increased computational cost, and larger block sizes, than PDAs working in the time-domain. Some algorithms operate directly on the time-frequency representation, and are based on analyzing trajectories of sinusoidal components in the spectrogram (sonogram) of the signal (Auger and Flandrin 1995; McAulay and Quatieri 1990; Baseville 1989; Marques and Almeida 1986). This approach is efficient, but requires the storing of a large amount of data for calculating spectrogram matrix. Its performance is limited in terms of real time applications, since it introduces significant delay to the calculated pitch. It is interesting also to analyze algorithms proposed by Beauchamp (1993a), and Brown (1992).

Alternatively, a signal might be analyzed with different time-frequency representations, as shown by Janer (1995) and by Kwong et al. (1992). Such algorithms are computationally expensive, resulting however in good performance of a period detection of the processed signal, with the assumption that pitch fluctuations are slow and the signal in each analyzed frame is nearly stationary.

Many algorithms were also engineered and implemented by the researchers' staff of the Multimedia Systems Department, GUT (Czyzewski et al 2002; Dziubinski and Kostek 2004; Szczerba and Czyzewski 2005). One of the examples of such algorithms is the algorithm of a time-frequency domain type, where rough pitch track estimation is performed using a block processing procedure, for relatively large blocks. The algorithm is based on analyzing spectrum peaks representing harmonics within the signal block. This will be presented later on. As an extension to this method, an algorithm for retrieving instantaneous pitch contour, working both: in time- and frequency-domains, based on pre-estimated rough pitch track is also presented in this Chapter.

Spectrum Peak Analysis Algorithm

The proposed pitch detection algorithm, a so-called Spectrum Peak Analysis (SPA), is based on analyzing peaks in the frequency-domain representing harmonics of a processed signal. In many cases, the spectrum representing an analyzed signal is difficult to deal with in terms of choosing peaks that represent pitch. Often the largest peak of the spectrum does not represent the fundamental frequency of the analyzed signal, i.e. it can represent one of the higher order harmonics, which happens in the case of trumpet sounds, for example. In addition, some of the lower order harmonics may not exist, or may be covered by noise. Difficulties with choosing an appropriate peak and establishing its relation to pitch are the most common problems in pitch determination based on spectrum analysis, and often cause octave errors in the estimation procedure. Proposed algorithms successfully deal with such situations, since their performance is based on the assumption that only a few harmonics exist (or are above the noise level). If only one harmonic exists - i.e. the analyzed signal is of the sinusoidal type, or other harmonics are covered by noise – it is assumed to represent the fundamental frequency of the input signal.

Estimating pitch contour is performed using a block processing, i.e., a signal is divided into blocks with equal widths, whereas overlap can be time varying. Each block is weighted by the Hann window.

Harmonic Peak Frequency Estimation

The first step of the estimation process in each block, is finding the maximum of the spectrum signal. Such maximum is assumed to be one of the harmonics, and it is easy to establish its coordinates in terms of frequency. The chosen peak is assumed to be at the Mth harmonic of the signal. In experiments, M equal to 20 seems to satisfy all tested sounds, however set-

ting M to any reasonable value is possible. The spectrum resolution is the natural limitation of this approach. It is assumed that the minimum distance d between peaks representing neighboring harmonics must be 4 samples. Therefore, if the detected maximum index is smaller than $M \cdot d$, M is automatically decreased by the algorithm to satisfy the formulated condition. In the case of low frequency signals, block size in the analysis must be suitably large to perform pitch tracking. The next step is calculating M possible fundamental frequencies, assuming that a chosen harmonic (the largest maximum of the spectrum signal) can be 1,2,…, or Mth harmonic of the analyzed sound:

$$F_{fund}[i] = \sum_{i=1}^{M} \frac{F_M}{i} \tag{3.20}$$

where F_{fund} denotes a vector of possible fundamental frequencies, and F_M is a frequency of a chosen (largest) harmonic.

The main concept of the engineered algorithm is to test a set of K harmonics related to vector $F_{fund,}$ that are most likely to be the peaks representing pitch. A value of K is limited by F_M in the following way:

$$K = floor(\frac{F_s}{M}) \tag{3.21}$$

where $floor(x)$ returns the largest integer value not greater than x, F_s is the sampling frequency.

Based on M, F_{fund} vector and K, matrix of frequencies used in analysis can be formed in the following way:

$$FAM(i, j) = \sum_{i=1}^{M} \sum_{j=1}^{K} F_{fund}[i] \cdot j \tag{3.22}$$

where FAM denotes a matrix containing frequencies of M harmonic sets.

If M is significantly larger than K, and most energy carrying harmonics are of higher order (the energy of the first K harmonics is significantly smaller than this of the higher order ones, for example K, $K+1, ..., 2 \cdot K$), it is better to choose a set of K consecutive harmonics representing the largest amount of energy. Therefore the frequency of the first harmonic in each set (each row of FAM) does not have to represent the fundamental frequency. Starting frequencies of chosen sets can be calculated in the following way:

$$H_{\max set}[j] = \sum_{i=1}^{K} EH_{(i+j)\cdot F_{fund}}, \qquad j = 0,...,L-1 \tag{3.23}$$

where:
H_{maxset} is a vector containing the energy of the K consecutive harmonics of the chosen set, $H_{maxset}[k]$ is the sum of the energy of these K harmonics: $k \cdot F_{fund}$, $(k+1) \cdot F_{fund}$, ..., $(k+K) \cdot F_{fund}$, EH_f is the energy of the harmonic with a frequency equal to f, and L is the dimension of H_{maxset} vector: $L = floor(\frac{F_s}{F_{fund}} - K)$

The first frequency of each set is based on the index representing the maximum value of $H_{maxset:}$ $F_{start}[m] = ind_{max}[m] \cdot F_{fund}[m]$ *for m = 1,..., M.*

Finally, modified *FAM* can be formed in the following way:

$$FAM(i,j) = \sum_{i=1}^{M} \sum_{j=1}^{K} F_{start}[i] + F_{fund}[i] \cdot (j-1) \tag{3.24}$$

Harmonic Peak Analysis

Each set of harmonics, represented by frequencies contained in each row of *FAM* is analyzed in order to evaluate whether it is most likely to be a set of peaks related to a fundamental frequency among the remaining *M-1* sets. This likelihood is represented by V, while V is calculated for each set in the following way:

$$V = \sum_{i=1}^{K} H_v[i] \tag{3.25}$$

where $H_v[i]$ denotes the value of a spectrum component for ith frequency for the analyzed set.

If the analyzed spectrum component is not a local maximum - left and right neighboring samples are not smaller than the one assigned to the local maximum - then it is set to 0. In addition, if local maxima of neighboring regions of the spectrum are found, H_v is decreased - the values of the found maxima are subtracted from H_v.

Neighboring regions of the spectrum surrounding the frequency F_{H_v}, representing H_v, are limited by the following frequencies:

$$F_L = F_{Hv} - \frac{F_{fund}}{2} \tag{3.26a}$$

$$F_R = F_{H_v} + \frac{F_{fund}}{2} \tag{3.26b}$$

where F_L , F_R are frequency boundaries of the spectrum regions surrounding F_{H_v}, *and* F_{fund} is the assumed fundamental frequency of the analyzed set.

The fundamental frequency, related to the largest V, is assumed to be the desired pitch of the analyzed signal. As can be observed in Figs. 3.12-3.14, there are three situations possible. In Fig. 3.12, the analyzed spectrum peak value is not a local maximum, therefore it is set to 0. In addition, there are local maxima detected in the surrounding regions, which subtracted from H_v give a negative value. It is clear that in this situation H_v is very unlikely to be a harmonic. Fig. 3.13 presents the situation where H_v is a local maximum, and the surrounding maxima, opposite to those from Fig. 3.14, have small values. There are neighboring harminics in analyzed regions, which is the case that pitch candidates are larger than real pitch. Fig. 3.13 presents a peak, with surrounding regions, that is most likely to be related to pitch.

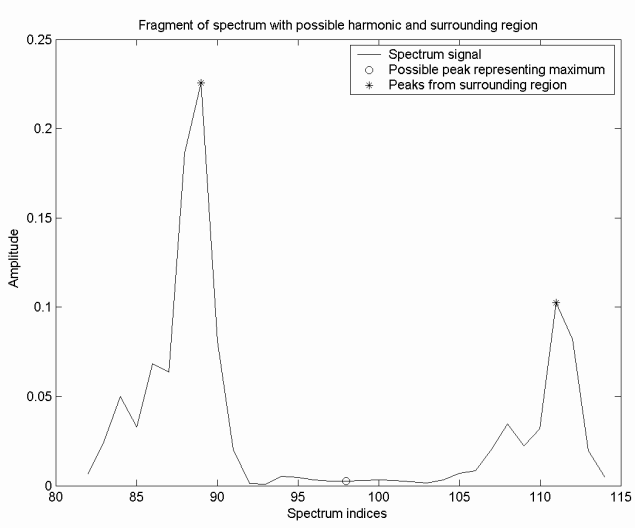


Fig. 3.12. Analysis of a possible harmonic peak and its surrounding region (the analyzed fundamental frequency is not related to peak frequency)

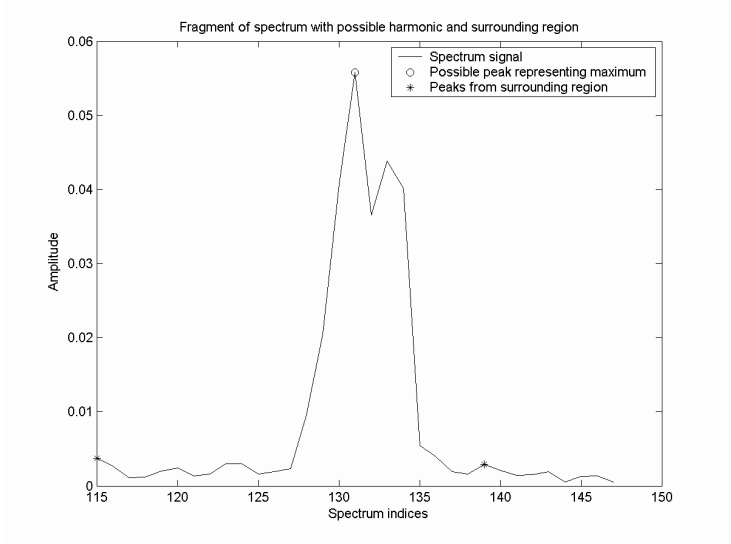


Fig. 3.13. Analysis of a possible harmonic peak and its surrounding region (the analyzed fundamental frequency is correctly related to peak frequency)

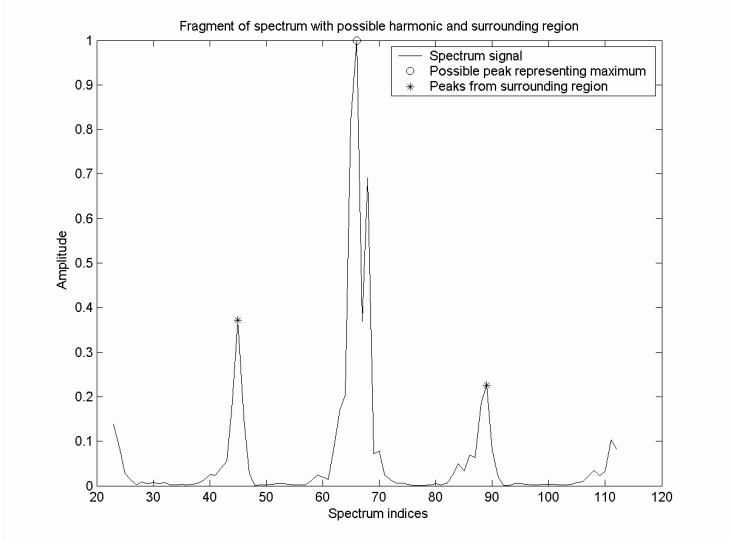


Fig. 3.14. Analysis of a possible harmonic peak and its surrounding region (the analyzed fundamental frequency is two times larger than pitch)

Time-domain Pitch Contour Correction

In some cases, transients of the analyzed fundamental frequency are two times larger than pitch analyzed instrument sounds may contain only or almost solely even harmonics, therefore pitch calculated for a block containing transient is one octave higher, than pitch calculated for blocks representing the steady-state of the sound. The human brain seems to ignore this fact, and for a listener the perceived pitch of the whole sound is in accordance with that of the steady-state. However, blocks containing transient, duplicated in time-domain, result in a sound with a pitch perceived as one octave higher. This observation calls for post-processing (Talkin 1995), i.e., time-domain pitch contour correction. Optimizing pitch tracks is relatively easy, since such problems are only encountered for transient parts of musical sounds and in most cases a pitch contour represents the fundamental frequency expected (perceived) for the instrument. In Fig. 3.15 one can observe that for an oboe sound, for one block in the transient phase, the estimated pitch is one octave higher than the pitch estimated for the steady-state, however, the overall pitch was recognized correctly for this sound.

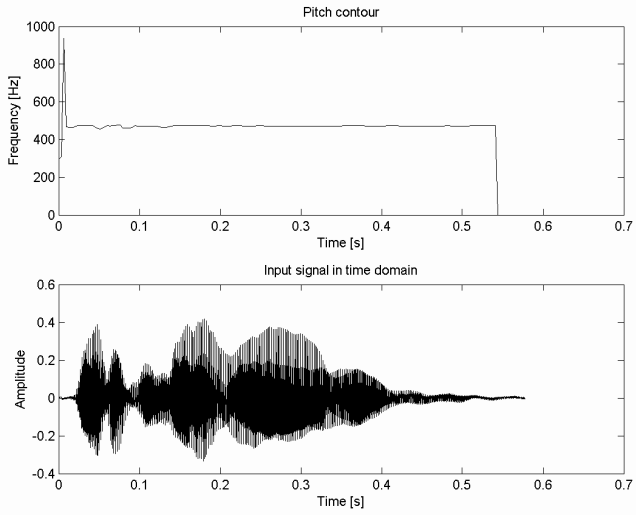


Fig. 3.15. Octave fluctuations of pitch in transient of oboe (*non legato*), based on SPA

Instantaneous Pitch Estimation

All known PDAs have limitations in estimating the pitch of the signal in terms of time resolution. This is due to the fact that some information from the signal has to be accumulated to perform pitch estimation. Time-domain based PDAs are limited by lengths of signal periods and in general, give higher resolution of the pitch contour in the time-domain, than spectrum (cepstrum) based PDAs. In spectrum (and cepstrum) related algorithms, the increase in block sizes results in higher resolution of the spectrum (cepstrum), but a calculated representation of the signal is based on a longer period of time and therefore the calculated pitch can be viewed as the average width of many periods contained in the analyzed frame.

A common solution for this problem is to increase overlap in the block processing procedure, which makes time resolution of the pitch track higher. The operation of increasing the overlap could be computationally inefficient and the estimated pitch contour could be understood as being a smoothed (by moving the average filter) and expected pitch track of the signal. Another way to enhance resolution of the estimation process is by using interpolation techniques on low resolution pitch contours. Interpolation may be performed based on a time-frequency representation of the signal (McAulay and Quatieri 1990; Stankovic and Katovnik 1998), or on a lower resolution pitch track calculated by a time-domain based algorithm (Medan et al 1988). In both cases, however, interpolation smoothes pitch contour, rather than introduces more precise information on the flowing pitch.

The method of directly retrieving a pitch track of the same time-domain resolution as this of the input signal is presented later on as a complementary method to SPA.

Data Segmentation

Direct retrieval of the pitch contour can be based on varying window sizes, i.e. block lengths can differ according to the deviation of a low resolution pitch track estimated with SPA – fluctuations of pitch within each analyzed frame should be less than 5%. To increase computational efficiency, window sizes should have lengths equal to the power of 2, since the FFT algorithm is involved. In addition, processed frames should be overlapping, since instantaneous pitch estimation (IPE) has lower performance on the edges of processed blocks – estimated pitch contour is affected by the Gibbs effect, and therefore the resulting pitch contours of overlapping blocks should be cross-faded to obtain a smooth instantaneous pitch contour for the whole signal.

Algorithm

The algorithm presented below is based on separating one of the harmonics of the signals and on a direct calculation of the instantaneous pitch, in which an inverted sinus function is used. In addition, methods of enhancing instantaneous pitch track (IPT) calculation are proposed.

The first stage of the procedure is calculating the spectrum of the signal (of a chosen block). A DCT algorithm is involved, since it provides a higher resolution of a spectrum. DCT can be expressed as:

$$x[n] = \omega[n]\sum_{k=1}^{N} x[n]\cos\frac{\pi(2n-1)(k-1)}{2N}, \qquad k = 1,...,N \tag{3.27}$$

where:

$$\omega[n] = \begin{cases} \dfrac{1}{\sqrt{N}}, & n = 1 \\ \sqrt{\dfrac{2}{N}}, & 2 \le n \le N \end{cases} \tag{3.28}$$

In addition, the input signal is weighted by the Hamming window. Since the pitch for a processed block is known (based on pre-calculated rough pitch contour), one of the spectrum peaks and its surrounding region, can be chosen to represent pitch fluctuation of one of the harmonics. Locating such a peak in the frequency-domain and calculating its representation in the time-domain is similar to the method proposed by d'Allessandro et al. (1995) and improved by Ahn et al. (1997). However, it is assumed here, that the region surrounding such a harmonic contains spectrum bins responsible for pitch fluctuations. This approach differs from both cited references, where the surrounding regions were treated as noise, and therefore the calculated harmonic estimate must be considered differently here.

It is best to choose the lowest harmonic, since its pitch fluctuations within the analyzed block are lowest. In some cases the lowest harmonic might contain very low energy, and might be strongly affected by the surrounding noise. In such a case it is more suitable to choose the lowest harmonic that contains a relatively large amount of signal energy. In experiments, one of the three lowest harmonics, containing the greatest energy and its surrounding spectrum bins was chosen to represent pitch fluctuations. The upper and lower frequency boundaries of the chosen spectrum fragment are related to pitch, and can be expressed as:

$$L_b = H_{freq} - P \cdot d \tag{3.29a}$$

$$H_b = H_{freq} + P \cdot d \tag{3.29b}$$

where:

L_b - lower boundary of a spectrum fragment surrounding a chosen harmonic peak,

H_b - higher boundary of a spectrum fragment surrounding a chosen harmonic peak.

P – average pitch of an analyzed block,

H_{freq} - frequency of a chosen harmonic peak,

d – experimentally set to 0.2, a deviation of pitch.

A time-domain representation of a chosen harmonic (H) was obtained by calculating inverse DCT of a chosen fragment of spectrum (other spectrum bins were zeroed). To decrease the Gibbs effect in the time-domain, the spectrum fragment was weighted by the Hann window (with the width equal to the spectrum fragment width). A calculated signal can be viewed as a sinusoidal signal with the time varying pitch and amplitude. The next step was to calculate a so-called analytic signal of the time representation of a chosen harmonic using the Hilbert transform. Absolute values of the analytic signal were treated as instantaneous amplitudes of the analyzed sinusoidal signal. Normalizing H by its instantaneous amplitude and calculating the arc sinus function resulted in $[-\frac{\pi}{2}, \frac{\pi}{2}]$ time vector. After unwrapping the time vector, the instantaneous pitch calculation can be expressed as:

$$P_c[n] = (t[n+1] - t[n]) \cdot \frac{f_s}{2\pi}, \qquad n = 1,...,N-1 \tag{3.30}$$

where P_c is an instantaneous pitch contour, t denotes an unwrapped time vector, f_s is a sampling frequency, and N is a size of an analyzed block.

The procedure of estimating IPT was presented in one of the author and her Ph.D. student's studies, therefore it will not be reviewed here. The examples of the experiments and their results will be presented later on.

3.2.4 Parametric Representation

Parametrization can be considered as a part of feature selection, the latter process understood as finding a subset of features, from the original set of pattern features, optimally according to the defined criterion (Swiniarski 2001). Creating a numerical representation of musical structures to be used in automatic classification systems requires defining a method of representing sound pitch, methods of representing time-scale and frequency properties, timbral characteristics, methods of representing other musical properties by feature vectors. The parametric approach allows one to describe the sound as a path through a multi-dimensional space of timbre. As presented in Chapter 2, musical sound timbre is a notion of features to be searched in the multidimensional space (Grey 1977). More dimensions can help to distinguish between particular instruments or musical instrument groups. One of the vital problems, still unsolved, is the relationship between sound descriptors and objectively derived parameters. Only a few parameters such as for example brightness have got their unquestioned interpretation - this subjective descriptor is related to a spectral centroid.

One can name such parameters, in both subjective and corresponding measures, as: pitch (frequency in Hz or barks), brightness (spectral centroid), tone/noise-like-quality (spectral flatness measure), attack asymmetry (skewness) or attack duration, overshoot or inharmonicity (log ratio of 1st harmonic to 2nd harmonic or more generally higher frequency harmonics to fundamental frequency ratio), vibrato (periodic fluctuation of pitch), tremolo (periodic change of sound level), nasality (formant positions if exist), synchronicity (delay of higher harmonics with relation to the fundamental during the attack), etc. that have dual interpretation. In addition there are parameters on the basis of which a distinction between musical instrument groups can be made. For example, skewness is a measure of data symmetry, or more precisely the lack of symmetry. A distribution is symmetric if it looks the same to the left and as to right of the center point. The skewness for a normal distribution is zero, and any symmetric data should have a skewness that aproaches zero. Negative values for the skewness indicate data that are skewed to the left (the left tail is heavier than the right tail), and positive values for the skewness indicate data that are skewed to the right. One can use such a statistical measure to describe the distribution of harmonics, which is different for woodwind and brass instruments.

The review of parameters shown below is based on the author's experiences with musical signal analysis, some of which may be found in litera-

ture. However, many of the presented parameters are derived from speech analysis, or/and are already standardized within the MPEG-7 framework.

In general, there are many approaches to feature vector extraction from musical sounds. Problems in signal processing involve time-dependent data for which exact replication is almost impossible. However, much of this time-dependent data arises from physical phenomena which can be considered as unchanging in their basic nature within periods of time. There are also parameters that are related to the time domain, but they are calculated on the basis of the frequency domain. Correlation parameters and parameters based on cepstral analysis may be included in this group. A specific model of sound production underlies some of the analysis methods (i.e. Linear-Prediction Coding (LPC), cepstral analysis methods, etc.). It is therefore necessary to have some kind of knowledge about the instrument that produces the signal. The results of the convolution between the excitation source and the resonance structure results in formants in the signal spectrum (De Poli et al 1991). However, most instruments have more than two acoustic systems coupled together, so the deconvolution of the excitation and the resonance systems is not easy. The spectral domain is also important for parameter derivation. Moreover, any study on musical sounds should take into account not only the physical way in which sounds are generated, but also the subsequent effect on the listener. In the latter case, some features of the perceptual model of the human hearing process, such as subjective loudness impression or masking effects, might be taken into account.

Another method to be mentioned is the analysis-by-synthesis approach. This approach in musical acoustics was actually introduced by Risset (De Poli et al 1991) in order to determine the most important sound parameters. In this case, the resynthesis of a sound is made on the basis of calculated parameters. For example, a harmonic-based representation of musical instrument tones for additive synthesis may be used as a sound parametrization. Although this data representation is usually very large, the principal component analysis can be used to transform such data into a smaller set of orthogonal vectors with a minimal loss of information (De Poli et al 1991). The analysis-by-synthesis method is also a way of verifying whether a chosen parameter is of good quality. If it is possible to resynthesize a sound on the basis of parameters, and it is still perceived as close to the natural one, then the parameters may be considered as appropriate.

It should be remembered that the choice of parameters and their number are crucial to the effectiveness of automatic classification processes.

Time Domain Representation

Generally, the ADSR model (see Fig. 3.16) may represent musical signal time domain characteristics, which is a linear approximation of the envelope of a musical sound. This time-domain representation is depicted as consecutive sound phases – Attack, Decay, Sustain and Release – that may be described in terms of their energy and time relationships.

The problem of locating the beginning of a sound is of importance, particularly in the sound automatic recognition process. Two time-domain measures - energy and the so-called zero-crossing rate are often used in the speech domain for the purpose of discriminating a speech utterance from background noise. For a signal $u=u(t)$, the zero-crossing function is defined as:

$$P(u,t)=\begin{cases}1\text{ - if there are signals } u(t) \text{ that fullfil}\\ \text{conditions (1), (2), and (3);}\\ 0\text{ - otherwise}\end{cases} \tag{3.31}$$

where:

$$\begin{aligned}&(1)\ u(t)\cdot u(t-\Delta t)<0\\ &(2)\ |u(t)|>\alpha \ \text{ and } \ |(t-\Delta t)|<\alpha\text{, where: } \alpha << \bar{u}\\ &(3)\ |u(t)|>\alpha \ \text{ for } t_0<t<t_0+\Delta t \text{ and } \Delta t=\frac{1}{f_{sampling}}\end{aligned} \tag{3.32}$$

Parameter α ($\alpha\neq 0$) is an assumed threshold.

The basic algorithms for the determination of a zero-crossing require a comparison of signs of pairs of successive samples in assumed time intervals. The distribution of such intervals is defined by the function $R(t)$:

$$R(t)=\sum_{j=1}^{J}\delta(t-t_j) \tag{3.33}$$

where: $\delta(t)$ - Dirac's delta, j=1, 2, ... J (J - number of zero-crossings) and t_j - time interval between the pair of (j-1) and j (in segment T), additionally:

$$T=\sum_{j=1}^{J}t_j \tag{3.34}$$

It should be remembered that the starting transients are the most important phase for the subjective recognition of musical sounds. It has been

shown in numerous experiments that when the attack phase is removed from a sound, it is no longer recognizable and, moreover, that some instrument sounds (trumpet and violin, for example) may not be distinguished from one another. In order to represent transient states, some parameters should be introduced. Krimphoff et al. introduced the rise time on a logarithmic scale (*LTM*), defined as (Krimphoff et al 1994):

$$LTM = \log(t_{\max} - t_{threshold}) \tag{3.35}$$

where $t_{\max}$ denotes time, when amplitude reaches the maximum value of the RMS, and $t_{threshold}$ is time corresponding to the minimum amplitude of signal threshold perception (see Fig. 3.16).

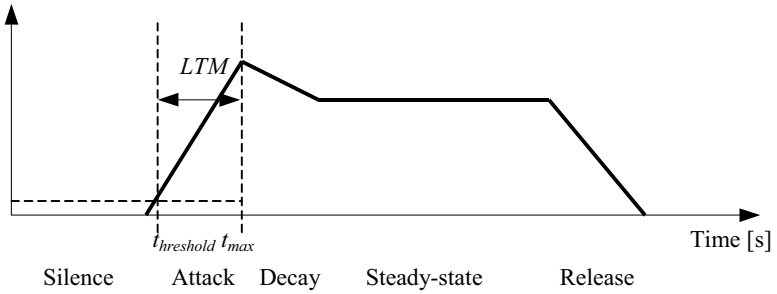


Fig. 3.16. Linear approximation of a musical signal envelope

The signal level versus time is defined as:

$$I(t) = a \int_{t-\frac{T}{2}}^{t+\frac{T}{2}} u^2(\tau) d\tau \tag{3.36}$$

where *T* is the width of the time window, and *a* is a normalization coefficient.

Another parameter represents the amplitude envelope (or instantaneous amplitude), described by the following expression:

$$O(t) = \sqrt{u^2(t) + \hat{u}^2(t)} \tag{3.37}$$

where $\hat{u}(t)$ denotes Hilbert Transform of the signal *u(t),* calculated as follows:

$$\hat{u}(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{u(\tau)}{t-\tau} d\tau \tag{3.38}$$

A parameter that is directly extracted from the time signal structure is the proposed transient midpoint, t_0 (see Fig. 3.17) (Kostek 1994, 1999).

The value of t_0 is calculated according to the formula:

$$t_0 = \frac{M_1}{M_0} = \frac{a+b}{2} \tag{3.39}$$

where M_1 is the first-order statistical moment:

$$M_1 = \int_{-\infty}^{\infty} t\,\mathrm{f}(t)dt = \left(b^2 - a^2\right)\frac{h}{2} \tag{3.40}$$

In order to normalize, the signal energy M_0 is calculated according to the following equation:

$$M_0 = \int_{-\infty}^{\infty} \mathrm{f}(t)dt = (b-a)\frac{h}{2} \tag{3.41}$$

where h is an energy increment versus time.

The envelope rising time may be found by the calculation of the second central moment:

$$t_{ris} = b - a = \sqrt{\frac{12M_2}{M_0}} \tag{3.42}$$

where:

$$M_2 = \int_{-\infty}^{\infty} (t-t_0)^2\,\mathrm{f}(t)dt = \frac{(b-a)^3 h}{12} = (b-a)^2\,\frac{M_0}{12} \tag{3.43}$$

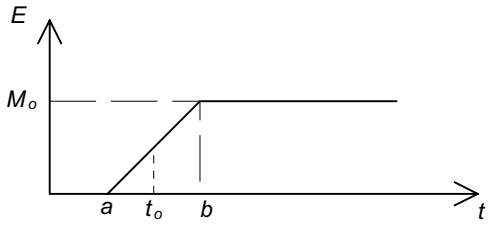


Fig. 3.17. Time envelope of the simplified transient model: a - transient starting point, b - transient ending point, M_0 - energy of the steady-state, t_0 - transient mid-point

There are two more phases that should be taken into account, namely the phase of energy decreasing from the local maximum and the subsequent phase of energy increasing from the local minimum to the energy of the steady-state.

Among others, ratio of time release ($t_{release}$) to sound duration (t) may be determined; however this parameter is susceptible to reverberation conditions.

$$P_{t2} = \frac{t_{release}}{t} \tag{3.44}$$

Also, velocity of sound release; this parameter is less susceptible to reverberation conditions than the parameter shown above:

$$rl = \frac{dx_{release}}{dt} \tag{3.45}$$

The essential factor that differentiates the ideal signal model from real sound recordings is the amplitude variation of the steady-state phase. As the amplitude of the musical signal varies with time, the signal energy provides a convenient representation that reflects these amplitude variations. For example, a difference between maximum (a_{MAX}) and minimum (a_{MIN}) amplitude may be determined:

$$D = \left| a_{MAX} - a_{MIN} \right| \tag{3.46}$$

Variances representing these fluctuations should be also considered, thus these parameters may be included in the feature vector.

Spectral Parameters

The feature vectors containing time domain parameters should be completed by adding the spectral properties. On the basis of the sound spec-

trum, many additional parameters may be determined. Moreover, as is seen from the above spectral estimation analysis, such methods may be used in the parametrization process, however, due to the high computational complexity, they make difficult an automatic analysis of musical sounds which is a real disadvantage while dealing with a musical sound database. It should be pointed out that the whole process starting from sound editing, through parametrization, and up to the classification process should be automatized. Additionally, parametric methods may cause uncontrolled loss of information. Therefore, in further analysis only parameters derived from the FFT-based analysis will discussed.

The spectrum components midpoint value f_m may be calculated using the following formula:

$$f_m = \frac{\int_0^{f_{\max}} f \cdot E(f) df}{\int_0^{f_{\max}} E(f) df} = r_f \frac{\sum_{i=1}^{I} i \cdot E_i}{\sum_{i=1}^{I} E_i} \tag{3.47}$$

where: r_f - parameter characterizing the resolution of the FFT analysis, E_i - energy of ith component for the frequency equal to r_f, f_{max} - upper limit of the analyzed frequency band, I - highest spectral component ($I \approx f_{max} / r_f$) (Kostek 1999).

Another parameter that is often used in the speech processing domain is the mth order spectral moment. It may be defined as follows (Tadeusiewicz 1988):

$$M(m) = \sum_{k}^{\infty} |G(k)| [f_k]^m \tag{3.48}$$

where: f_k - is a center frequency of the kth frame used in the spectral analysis. Values of f_k may be calculated on the basis of Eq. (3.49), in which the resolution (Δf) of spectral analysis is used:

$$f_k = (k-1)\Delta f + \frac{\Delta f}{2} \tag{3.49}$$

The parameter defined by Eq. (3.48) may be interpreted physically. For example, on the basis of the 0-order spectral moment, the energy concentration in the low frequencies may be exposed. Also, this parameter is often used as a normalization coefficient for the higher order spectral mo-

ments. On the other hand, the 1st order spectral moment may be interpreted as spectral centroid coefficients. $G(k)$ in Eq. (3.48) are dependent on the window function that was applied to the analysis. In the case where the spectral domain is represented by components of amplitudes A_k and frequencies which are *n*th multiplies of the fundamental, then the above shown relationship (3.49) should be modified according to Eq. (3.50). Therefore, the *m*th moment may be calculated as follows:

$$M(m) = \sum_{k=1}^{n} A_k (k)^m \tag{3.50}$$

and the spectral centroid (*Brightness*) may be defined as follows:

$$B = \sum_{n=1}^{N} n \cdot A_n / \sum_{n=1}^{N} A_n \tag{3.51}$$

where: A_n - amplitude of the *n*th harmonic, N - total number of harmonics.

Other spectral moments are also valuable, for example 3rd and 4th order.

There are other parameters which describe the shape of the spectrum in the steady-state phase, such as the even (h_{ev}) and odd (h_{odd}) harmonic content in the signal spectrum:

$$h_{ev} = \sqrt{\frac{A_2^2 + A_2^2 + A_2^2 + \ldots}{A_1^2 + A_2^2 + A_3^2 + \ldots}} = \frac{\sqrt{\sum_{k=1}^{M} A_{2k}^2}}{\sqrt{\sum_{n=1}^{N} A_n^2}} \tag{3.52}$$

M = entier(N/2);

and contents of odd harmonics in the spectrum, excluding the fundamental:

$$h_{odd} = \sqrt{\frac{A_3^2 + A_5^2 + A_7^2 + \ldots}{A_1^2 + A_2^2 + A_3^2 + \ldots}} = \frac{\sqrt{\sum_{k=1}^{L} A_{2k-1}^2}}{\sqrt{\sum_{n=1}^{N} A_n^2}} \tag{3.53}$$

L = entier(N/2 + 1).

where: A_n, N - as before.

Another parameter derived from the frequency domain which is often used for the purpose of estimation of auditory masking effects seems of importance (Zwicker and Zwicker 1991), namely the *Spectral Flatness Measure* (*SFM*). Since audio signal varies in successive frames of the analysis, the *SFM* parameter may thus be used as a measure of the tonal (clear maxima) or noiselike (flat spectrum) character of the signal, expressed as the ratio of the geometric to arithmetic mean of the power density function, defined as follows (Johnston 1988; Zwicker and Zwicker 1991):

$$SFM = 10\log_{10}\left(\frac{\left[\prod_{k=1}^{N/2} P(e^{j\frac{2\pi k}{N}})\right]^{1/\frac{N}{2}}}{\frac{1}{N/2}\sum_{k=1}^{N/2} P(e^{j\frac{2\pi k}{N}})}\right) i \tag{3.54}$$

where: $P(e^{j\frac{2\pi k}{N}})$ is the spectral power density function calculated on the basis of the N-point Fast Fourier Transform Algorithm.

On the basis of the *SFM* value, an additional parameter is formulated, namely a coefficient of tonality a that is expressed as:

$$a = \min\left(\frac{SFM}{SFM_{\max}}, 1\right) \tag{3.55}$$

where: a = 1 for $SFM=SFM_{max}$=-60dB (sine wave), and a = 0 for SFM=0dB (white-noise signal).

Formants are parameters widely used in speech analysis which indicate local maxima of the spectrum. It is obvious that their physical interpretation in musical acoustics corresponds to resonances of the instrument body. Precise tracking of the formant frequency is not easy. However, using amplitudes of discrete spectrum A_1, A_2, A_3, A_4 and corresponding frequencies f_1, f_2, f_3, f_4 it is possible to calculate the approximate formant frequency as $\hat{F}$ or $\tilde{F}$ (Tadeusiewicz 1988):

$$\hat{F} = \frac{A_1 f_1 + A_2 f_2 + A_3 f_3}{A_1 + A_2 + A_3} \tag{3.56}$$

$$\tilde{F} = \frac{A_2 f_2 + A_3 f_3 + A_4 f_4}{A_2 + A_3 + A_4} \tag{3.57}$$

For a simplified formant tracking algorithm, the following assumptions are to be made: the formant is located among the neighboring components: (*k*-*p*), (*k*+*p*), if the following conditions are fulfilled (Kostek 1995a):

1. values of component amplitudes are bigger than the assumed threshold value $A_{threshold}$:

$$(A(k) \geq A(k-p) \wedge A(k) \geq A(k+p)) \geq A_{threshold} \tag{3.58}$$

where: *p* defines the demanded width of the formant.

2. *df*, defined as the difference between the spectral centroid and geometrical center, taken with the minus sign, is bigger than the assumed threshold $df_{threshold}$.

The presented algorithm was applied by the author in order to extract formant frequencies in musical sounds (Kostek 1995a). The threshold value of $A_{threshold}$ may be expressed in terms of the amplitude mean value or of the RMS value, as defined below:

$$RMS = \sqrt{\sum_{k=1}^{n} A_k^2} \tag{3.59}$$

It should be mentioned that formants, i.e. enhancements of harmonics in certain fixed frequency intervals, remain invariable within the chromatic scale of instrument, whereas spectra of individual tones may vary considerably from one note to another. Thus this feature is specific for a given instrument.

Another criterion (*IRR*) introduced by Krimphoff and al. corresponds to the standard deviation of time-averaged harmonic amplitudes from a spectral envelope, and is called ‘spectral flux’ or ‘spectral fine structure’ (Krimphoff et al 1994):

$$IRR = 20\log\left(\sum_{k=2}^{n-1}\left|A_k - \frac{A_{k+1} + A_k + A_{k-1}}{3}\right|\right) \tag{3.60}$$

In the literature, an approach to the estimation of the sound spectral domain based on polynomials may be also found. This approach seems to be especially justified in the case of a rich sound spectrum. The applied approximation is based on minimizing the mean-square error in the range of the analyzed spectrum by using the following proposed relation (Kaczmarek et al 1998; Kostek 1995a, 1995c):

$$E = \sqrt{\sum_{i=1}^{N} (20 \cdot \log_{10} A(i) - W_l(\log_2 i))^2} \tag{3.61}$$

while:

$$W_l(\log_2 i) = \sum_{j=0}^{l} a_j \cdot (\log_2 i)^j \tag{3.62}$$

where:

E - mean-square error,
i - number of the harmonic, i=1, 2,...N,
N - number of the highest harmonic,
$A(i)$ - value of the amplitude of ith component,
$W_l(\log_2 i)$ - value of the polynomial for ith component,
a_j - jth term of the polynomial,
j - number of the consecutive term of the polynomial,
l - order of the polynomial.

Computations which minimize the error are performed by consecutive substitution of the order of the polynomial (l=1,2 ... etc.), successively obtaining coefficients a_1, a_2, a_3, ... Based on formula (3.61), an approximation is performed in the spectrum domain, presented in the *log/log* scale, which causes the consecutively computed coefficients a_j to have units respectively *dB/octave*, *dB/octave*^{2}, *dB/octave*^{3}, etc. These coefficients have a clear physical interpretation, e.g. the first defines the decay of higher harmonics in the spectrum, whereas the second indicates a gain or a loss of the middle part of the spectrum in relation to its lower or higher parts. By raising the approximation order, more coefficients are obtained which describe more precisely the spectrum of the sound. The minimum number of polynomial coefficients approximating the envelope spectrum may be determined in listening tests.

An illustration of such an approach is shown in Fig. 3.18. It was proved based both on the mean-square error optimization and listening tests that

the 5th order of the approximating polynomial may be considered as sufficient in the cases of both shown instruments.

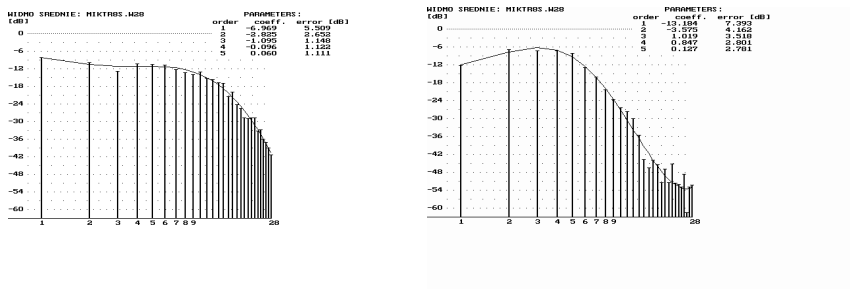


Fig. 3.18. Sound spectra approximated by the 5th order polynomial

Wavelet-Based Parameters

In order to define parameters that may be derived from the wavelet-based transform some extensive experiments were performed at the Multimedia Systems Department, GUT (Kostek and Czyzewski 2000; Kostek and Czyzewski 2001a; Kostek and Czyzewski 2001b). Several filters such as proposed by Daubechies, Coifman, Haar, Meyer, Shannon, etc. were used in analyses and their order was varied from 2 up to 8. It was found that Daubechies filters are sufficiently effective and the computational load was the lowest in this case (Kostek and Czyzewski 2001b).

In order to visualize differences in analyses obtained using FFT and wavelet transform, two exemplary analyses will be discussed. In Fig. 3.19 the FFT sonogram and time-frequency analysis (Fig. 3.20) are presented for a violin sound (A4, *non_legato*, *forte*). In the case of Fig. 3.20 a rectangle in the so-called phase space is associated with each wavelet basis function (MATHEMATICA 1996). The larger the absolute value of the corresponding wavelet, the darker a rectangle. In order to analyze the starting transient of the exemplary violin sound the number of samples was assigned to 2048 (46.44 ms), because the steady-state begins approximately at 58 ms. Since the analyzing windows in the implemented wavelet algorithms in the MATHEMATICA system are octave-based (MATHEMATICA 1996), thus this was an optimum choice of the window size. In both plots shown in Fig. 3.19 and in Fig. 3.20 the increase of higher frequency harmonics energy with time is visible.

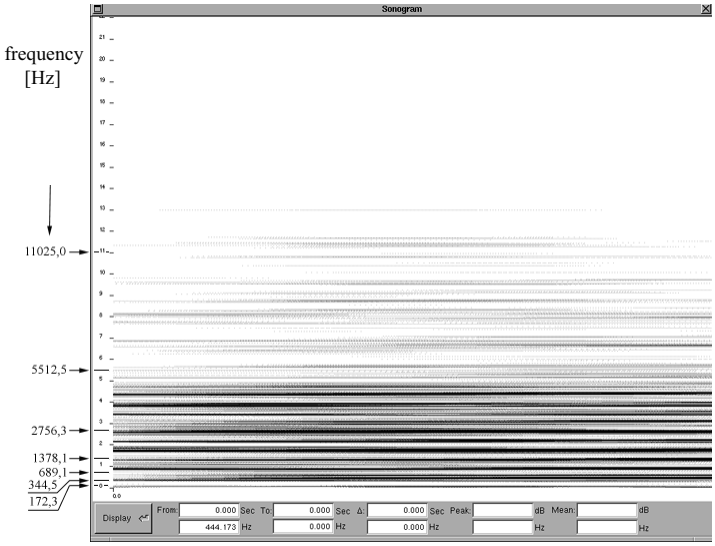


Fig. 3.19. FFT sonogram of violin sound (A4)

Looking at the wavelet analyses one should observe which specific subband is the most significant energetically. It should be remembered that wavelet subbands could contain more than only one sound harmonics. This would allow associating the amount of energy that is related to low, mid and high frequencies. Secondly, it is interesting when the summed up consecutive wavelet coefficients within selected subbands would attain a certain energy threshold. The algorithm allowing for finding this time instance will return the number of the sample (or time in ms) corresponding to the normalized energy threshold (Kostek and Czyzewski 2000, 2001b). This parameter may differentiate the articulation features between musical sounds.

The cumulative energy $E_c(n)$ is defined as squared modulus of the corresponding coefficient c_i that represents the original data (MATHEMATICA 1996):

$$E_c(n) = \sum_{i=1}^{n} \left| c_i \right|^2 , \qquad \left| c_i \right| \geq \left| c_{i+1} \right| \tag{3.63}$$

Taking into account this parameter it is possible to perform the inverse wavelet transform by retaining only significant coefficients. It can be seen that in the case of a trumpet sound (see Fig. 3.21a), fewer coefficients should be retained for performing the inverse wavelet transform than in the case of a violin sound (Fig. 3.21b). It should be noticed that approximately 70% of energy are concentrated in the first 40 coefficients. Among others, such a parameter can be used as one that provides discrimination between instruments.

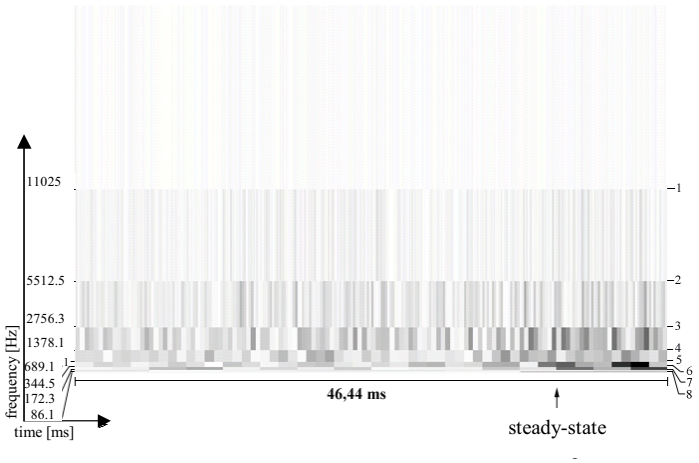


Fig. 3.20. Time-frequency analysis of A4 violin sound (the vertical scale corresponds to the frequency partition in the case of sampling frequency equal to 44.100 Hz, the horizontal scale is expressed in time [ms] that corresponds to the number of samples taken to analysis)

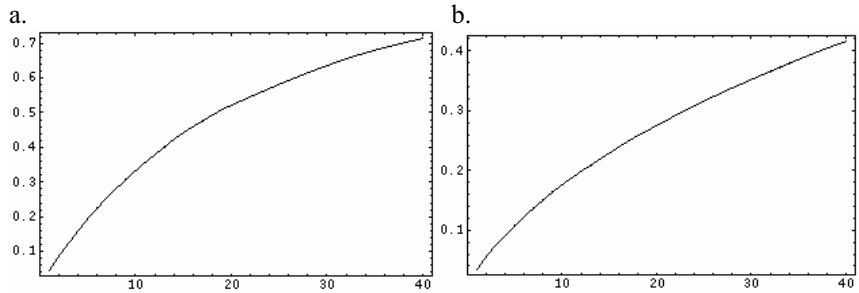


Fig. 3.21. Wavelet analyses of the trumpet (a) and violin (b) sound (*non_legato*, *forte*) – cumulative energy versus sample packet number

Several other parameters can be determined on the basis of the experiments performed. They were calculated for the Daubechies filter of order 2 (number of samples in the analysis frame was equal to 2048) as:

- E_n – partial energy parameters,

where:

$$E_n = \frac{E_i}{E_{total}} \tag{3.64}$$

$$E_i = \left(\sum_{k=1}^{K} c_k \right) \cdot w_i \tag{3.65}$$

where:

c_k – consecutive wavelet coefficients

w_i – weight applied in order to normalize E_i (resulted from different number of coefficients in wavelet spectrum bands)

$E_i = E_1 \ldots, E_{10}$ – energy computed for the wavelet spectrum bands normalized to the overall energy E_{total} of the parameterized frame corresponding to the starting transient, where:

i=1 – energy in the frequency band 21.53-43.066Hz,

i=2 - energy in the frequency band 43.066-86.13Hz,

i=3 - energy in the frequency band 86.1-172.26Hz,

i=4 - energy in the frequency band 172.26-344.53Hz,

i=5 - energy in the frequency band 344.53-689.06Hz,

i=6 - energy in the frequency band 689.06-1378.125Hz,

i=7 - energy in the frequency band 1378.125-2756.26Hz,

i=8 - energy in the frequency band 2756.26-5512.5kHz,

i=9 – energy in the frequency band 5512.5-11025 Hz,

i=10 – energy in the frequency band 11025-22050Hz,

– number of the sample that corresponds to the normalized energy threshold $E_{threshold}$ calculated for each kth subband $t_{threshold}(k) = t_{th1} \ldots\ldots t_{th10}$ (Kostek and Czyzewski 2000), where:

$$E_{threshold} = \alpha \cdot E_{total}, \; 0<\alpha<1 \tag{3.66}$$

and α – coefficient assigned arbitrarily,

– rising time of starting transient t_{start}

In Fig. 3.22, sample results of the wavelet-based feature extraction (E_n) are shown for some chosen instruments. In all cases a frame consisting of 2048 sound samples was analyzed. In Fig. 3.22 energy values are presented for ten wavelet spectrum sub-bands. The whole instrument range

can be seen within each sub-band. Left side lines within each sub-band correspond to the lowest sounds, whereas the right side lines to the highest ones. It can be observed that energy distribution pattern within the wavelet spectrum sub-bands differentiates between trumpet and a violin. Although this parameter is sensitive both to type of instrument and sound pitch, it is also, in a way, characteristic for wind and string instruments.

The rising time of the starting transient was defined as a fragment between the silence and the moment in which the signal would attain 75% of its maximum energy.

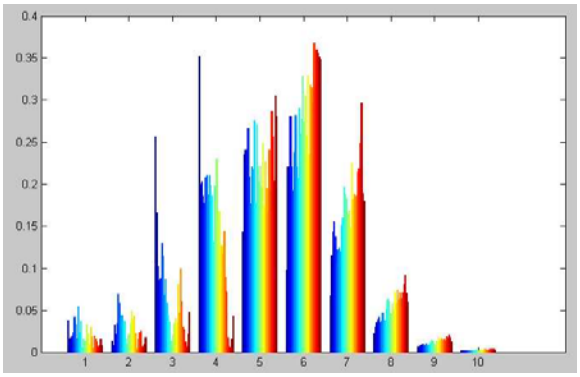
Additionally, the end-point of the transient - t_{end} can be determined according to the following condition:

$$\left| \max_{i=i_0,\ldots,i_0+T} s[i] - \max_{i=i_0+T+1,\ldots,i_0+2T} s[i] \right| < 0.1 \cdot \max_{i=i_0,\ldots,i_0+2T} s[i] \tag{3.67}$$

where: T – observation period expressed in samples;

– cumulative energy - $\underline{E_c}$ is conditionally determined, when the maximum relative error of the energy change between the original signal and the reconstructed one on the basis of the retained wavelet coefficients is less than 20%.

a.



(Legend to Fig. 3.22; see next page)

b.

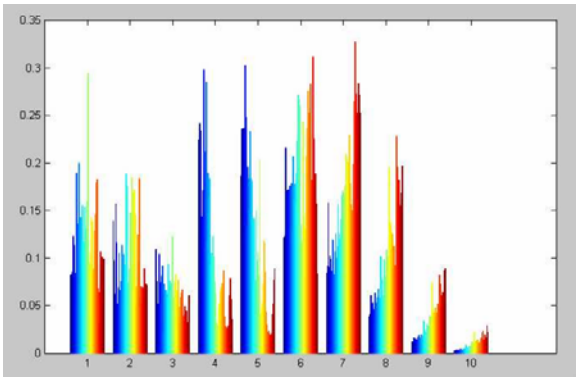


Fig. 3.22. Values of the E_n parameter for selected instruments: a – trumpet, b – violin

In the experiments several time-related parameters were also explored. Two of them, the most significant from the statistical point of view, were included in the feature vector. If there are certain parameters that allow for easier distinguishing between particular instrument classes and others that will do the same for other classes, it is thus possible to design a system consisting of a few preprocessing blocks that will first separate for example groups of parameters. Thus two additional relations are defined in the wavelet discrete-time domain. They are presented below:

$$e = \left(\frac{\sum_{n=n_0}^{N} \left(|c_n| > 0.2 \cdot |c_{\max}| \right)}{512 - n_0} \right) \cdot \frac{1}{s} \tag{3.68}$$

where e is time-related parameter allowing for characterization of the wavelet pattern, calculated for each wavelet spectrum subband, c_n are wavelet coefficients, n_o is the first wavelet coefficient that exceeds the assigned threshold, and s refers to sound pitch.

$$f = \mathrm{var}(f'[|c_n|]) \tag{3.69}$$

where f is variance of the first derivative of the absolute value of the wavelet coefficient sequence.

The parameter *e* from expression (3.68) refers to the number of coefficients that have exceeded the given threshold. This threshold helps to differentiate between 'tone-like' and 'noise-like' characteristics of the wavelet spectrum. The value of such a threshold was assigned experimentally to 0.2. It then returns the associated sample number. It takes approximately 180 samples for a trumpet sound, and 220 samples for a clarinet to attain this threshold. The meaning of the parameter *f* is the variance estimation of the derivative of the sequence of coefficients.

Other Parameters

It is convenient to correlate time-related properties with those of the frequency-domain (Pollard and Jansson 1982). In the *Tristimulus* method, loudness values measured at 5ms intervals are converted into three coordinates, based on the loudness of (1) the fundamental (N_1), (2) the group containing partials from 2 to 4 (N_2), and (3) the group containing partials from 5 to N (N_3), where N is the highest significant partial. The values of (N_2) and (N_3), are calculated according to the formula:

$$N_{2(3)} = 0.85N_{\max} + 0.15N_i \tag{3.70}$$

where: N_{max} - component having the maximum loudness within the given group of harmonics.

Then, parameters *x, y, z* are derived from the following formulae:

$$x = \frac{N_3}{N}; \quad y = \frac{N_2}{N}; \quad z = \frac{N_1}{N} \tag{3.71}$$

where:

$$N = \sum_{i=1}^{3} N_i \tag{3.72}$$

This procedure allows a simple graph to be drawn that shows the time-dependent behavior of the starting transients with relation to the steady-state.

Furthermore, harmonic energy or amplitude values may be taken into account instead of loudness for classification purposes (Kostek 1995a, 1995b; Kostek and Wieczorkowska 1996). Therefore, three parameters are extracted for the above defined spectrum subbands, namely the first - T_1, second - T_2, and third - T_3, modified *Tristimulus* parameters according to the formula:

- the first modified *Tristimulus* parameter

$$T_1 = A_1 / \sum_{n=1}^{N} A_n^2 \tag{3.73}$$

where: A_n, N - defined as before.

- the second modified *Tristimulus* parameter:

$$T_2 = \sum_{n=2}^{4} A_n^2 / \sum_{n=1}^{N} A_n^2 \tag{3.74}$$

- the third modified *Tristimulus* parameter:

$$T_3 = \sum_{n=5}^{N} A_n^2 / \sum_{n=1}^{N} A_n^2 \tag{3.75}$$

Additionally, the following condition is to be imposed to the above defined parameters:

$$T_1 + T_2 + T_3 = 1 \tag{3.76}$$

As most of the presented parameters do not have stable values within the chromatic scale of an instrument, the applicability of other criteria has been verified, such as the mel-cepstrum coefficients (*MCC*) defined by the following expression (Kostek 1995b):

$$W_c[k] = \sum_{i=1}^{n} E_i \cos(\frac{\pi}{n}(i - 0.5) \cdot k) \tag{3.77}$$

where: $W_c[k]$ - kth cepstrum coefficient, E_i - energy of ith harmonic expressed in [dB].

Such parameters as mel-cepstrum were used in many studies on musical sounds classification, examples of which may be (Brown 1999; Cosi et al 1994b).

Also, parameters that are related to the frequency of the nth harmonic – normalized frequency deviation and inharmonicity, were examined in literature (Beauchamp 1993b). The first factor is defined in the following formula:

$$\frac{\Delta f_n(t)}{nf_1} = \frac{f_n(t)}{nf_1} - 1 \tag{3.78}$$

where: f_n - frequency of nth harmonic, f_1 - fundamental frequency;

The inharmonicity factor describing the degree to which a sound is not perfectly harmonic is given below:

$$inh = \frac{f_n(t) - nf_c(t)}{nf_1} \tag{3.79}$$

where:

$$\frac{\Delta f_c(t)}{f_1} = \frac{f_c(t)}{f_1} - 1 = \sum_{k=1}^{5} A_k(t)(\Delta f_n / nf_1) / \sum_{1}^{5} A_k(t) \tag{3.80}$$

As seen from the above equation, an additional parameter, called the composite weighted-average frequency deviation, is defined. This is because it often happens in practice that the fundamental is much weaker than other harmonics. Therefore, the inharmonicity factor is determined for the five lowest spectrum partials as a frequency centroid (Beauchamp 1993b).

A convenient way to display certain properties of a signal is by using its statistical representation (Rabiner and Schafer 1978). For this purpose, autocorrelation (r_{An},r_{Fn}) and cross-correlation functions (r_{Amn},r_{Fmn}) are often defined (Ando and Yamaguchi 1993):

$$r_{An}(k) = \left[(M-k)\sigma_{An}^2\right]^{-1} \sum_{r=0}^{M-k-1} A_n(r) \cdot A_n(k+r) \tag{3.81}$$

$$r_{Fn}(k) = \left[(M-k)\sigma_{Fn}^2\right]^{-1} \sum_{r=0}^{M-k-1} \Delta F_n(r) \cdot \Delta F_n(k+r) \tag{3.82}$$

where: $k=0,1,...,M/2$, σ_{An},σ_{Fn} are standard deviations for the signal amplitude and frequency, respectively, and k is the time lag, which has a maximum value of $M/2$,

and:

$$r_{Amn}(k) = \left[(M-k)\sigma_{Am}^2\sigma_{An}^2\right]^{-1} \sum_{r=0}^{M-k-1} A_m(r) \cdot A_n(k+r) \tag{3.83}$$

$$r_{Fmn}(k) = \left[(M-k)\sigma_{Fm}^2\sigma_{Fn}^2\right]^{-1} \sum_{r=0}^{M-k-1} \Delta F_m(r) \cdot \Delta F_n(k+r) \tag{3.84}$$

where: σ_{Am}, σ_{An} and σ_{Fm}, σ_{Fn} are standard deviations between the nth and mth amplitudes and frequencies of signal harmonics, respectively (Ando and Yamaguchi 1993).

These functions provide information on the relationships between signal amplitudes and frequencies and are very useful in determining the signal periodicity.

There are more parameters that may be derived using various approaches to the musical signal analysis, such as fractal dimension (based on fractal interpolation of the spectrum envelope) (Monro 1995; Schroeder 1989). Fractal interpolation provides a new technique for generating sounds, thus defining it as a method of synthesis (Monro 1995). It produces functions in consecutive iterations that may be described on the basis of given points and a number reflecting the displacement of each line segment of the interpolating function. Suppose that the starting points in this method are (x_i, y_i) for i=0, 1,...,N and the displacements are d_i for i=1,...,N. In the case where the points $(x_1,...x_i,...x_N)$ are equally spaced and the original points and displacements do not lie in a straight line, the fractal dimension is given by the formula (Monro 1995):

$$D = 1 + \frac{\log\left(\sum_{i-1}^{N} |d_i|\right)}{\log(N)} \tag{3.85}$$

Apart from parameters presented, yet another parameter extracted from a single frame of the steady state of the audio signal may be determined.

Pitch (P) – expresses pitch of the sound according to MIDI standard:

$$P = 69 + 12 \cdot \log_2\left(\frac{f_0}{440}\right) \tag{3.86}$$

where f_0 is the fundamental frequency of audio signal. Sound pitch is denoted as KeyNum.

3.2.5 MPEG-7 Standard-Based Parameters

MPEG-7 Objectives

According to MPEG-7 Web home page, the MPEG-7 standard is understood as "Multimedia Content Description Interface", which enables to attach metadata to multimedia

(http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm). A huge amount of digitized audiovisual information is available in databases and archives, and in the Internet. The value of information depends on how well the data can be managed in terms of automatic retrieval, access, translation, conversion, filtration, etc.

Audiovisual data content that has MPEG-7 descriptions may include still pictures, graphics, 3D models, audio and speech signals, videos, and the information how these elements are combined in a multimedia presentation, so-called scenarios. MPEG-7 descriptions do not depend on the ways the described content is coded or stored. It is possible to create an MPEG-7 description of an analogue movie or of a picture that is printed on paper, in the same way as the one of digitized content (http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm).

The standard allows at least two types of levels of discrimination in its descriptions. The level of abstraction is related to the way the features can be extracted: many low-level features can be extracted fully automatically, whereas high level features need much more human interaction. Apart from a description of what is depicted in content, other types of information about the multimedia data, such as: form, conditions for accessing the material, classification, links to other relevant material, and context, are inquired.

The main elements of the MPEG-7 standard are (http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm):

- Description Tools: Descriptors (D), that define the syntax and the semantics of each feature (metadata element) and Description Schemes (DS), that specify the structure and semantics of the relationships between their components, which may be both Descriptors and Description Schemes
- Description Definition Language (DDL) used to define the syntax of the MPEG-7 Description Tools (Hunter 2001), to allow the creation of new Description Schemes and Descriptors, and to allow the extension and the modification of existing Description Schemes.

What is of the utmost importance, the MPEG-7 standard addresses many different applications in many different environments, which means that both tools and descriptors should be flexible and easily extensible. In addition, some interoperability with other metadata standards is already envisioned (http://www.w3.org/Metadata/).

MPEG-7 Standard-Based Parameters

The MPEG-7 standard refers to metadata information contained in the Internet archives. This notion is very often applied to the value-added information created to describe and track objects, allowing access to them. (http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm).

In this context descriptors that are well-defined provide better computing, improved user interface and data management. In the context of the MPEG-7 standard all information of higher level is defined as textual information on audio, such as titles of songs, signers' names, composers' names, the duration of music excerpt, etc. One should keep in mind that music can be described in a number of ways and that musical sounds include polyphonic sounds and human voice sounds (speech and singing). A musical signal, music, scores (graphical form), MIDI code or a verbal description, each comes as a different representation. Provided within the MPEG-7 standard, are also low-level descriptors for musical data, organized in groups of parameters, such as Timbral Temporal, Basic Spectral, Basic, Timbral Spectral, Spectral Basis, Signal Parameters (see Fig. 3.23) (http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm). A so-called Audio Framework that contains all these parameter groups includes 17 vector and scalar quantities. They represent: log(attack time), temporal centroid, audio spectrum envelope, audio spectrum centroid, audio spectrum spread, audio spectrum flatness, audio waveform and power, harmonic spectral centroid, harmonic spectral deviation, harmonic spectral spread, harmonic spectral variation, a spectral centroid, audio spectrum basis, audio spectrum projection, audio harmonicity and audio fundamental frequency (http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm). These low-level descriptors provide information for higher-level applications, such as: sound recognition, musical instrument timbre similarity, melody and melodic contour recognition, robust audio matching and spoken content recognition. It can easily be observed that these low-level descriptors are more data- than human-oriented. This is because the idea behind this standard is to have data defined and linked in such a way as to be able to use it for more effective automatic discovery, integration, and re-use in various applications. The most ambitious task is, however, to provide seamless meaning to low- and high-level descriptors. In such a way data can be processed and shared by both systems and people.

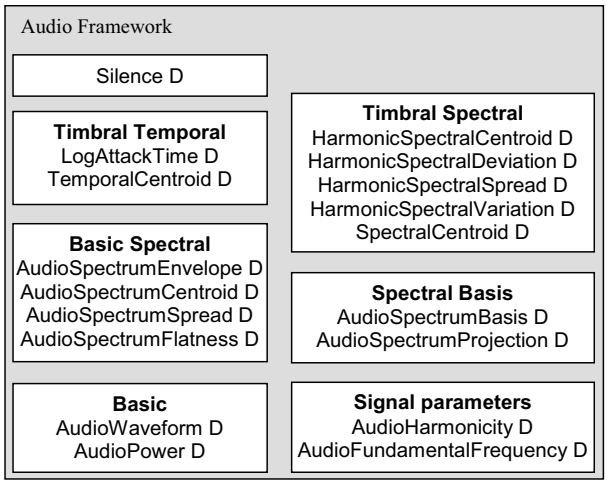


Fig. 3.23. MPEG-7 Audio Framework parameters

The majority of the analyzed sound descriptors are defined nowadays within the MPEG-7 standard framework (http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm). They may be divided into two groups: Audio Spectrum Descriptors and Timbre Descriptors. Descriptors of the first group are derived directly from the audio spectrum, whilst the descriptors of the second group are mostly based on the positions and the amplitudes of harmonic peaks.

Below, the practical implementation of the MPEG-7 standard parameters is shown. The extraction of all MPEG-7-based spectral descriptors involves a sliding window FFT analysis of the audio signal. The Hamming analysis window of the length of 30 ms has been chosen. The length between two adjacent windows is 10 ms, which means that 66% of the current window overlaps the previous window. Subsequently, FFT is performed in each window. A spectral descriptor is calculated separately in every spectral frame. Finally, the spectral descriptor is described by a two-dimension vector containing mean and deviation values of the descriptor in every spectral frame.

Definitions of the MPEG-7-based descriptors are as follows (http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm):

Audio Spectrum Descriptors:

- Audio Spectrum Envelope (*ASE*) – describes the short-term power spectrum of the waveform as a time series of spectra with logarithmic frequency axis. According to the MPEG-7 recommendation, the spectrum consists of one coefficient representing power between 0 Hz and

62.5 Hz, a series of coefficients representing power in ¼-octave resolution sized bands between 62.5 Hz and 16 kHz, and a coefficient representing the power beyond 16 kHz. This results in 34 coefficients for each spectral frame. The mean values and variances of each coefficient over time are denoted as $ASE_1 \ldots ASE_{34}$ and $ASEv_1 \ldots ASEv_{34}$, respectively. In Fig. 3.24 a presentation of spectrum subbands is shown. In addition, this function returns the conversion of linear to log bands (see Fig. 3.25 for conversion from linear to logarithmic scale).

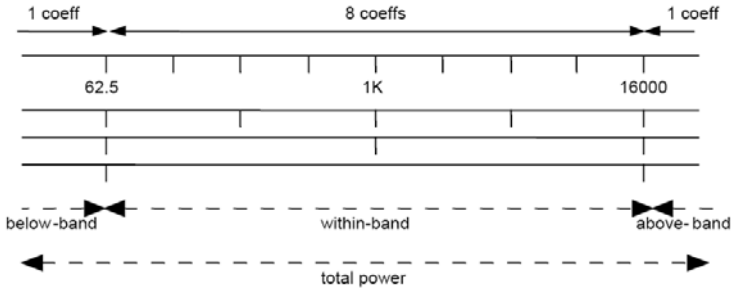


Fig. 3.24. Structure of the AudioSpectrumEnvelope Descriptor - presentation of spectrum subbands

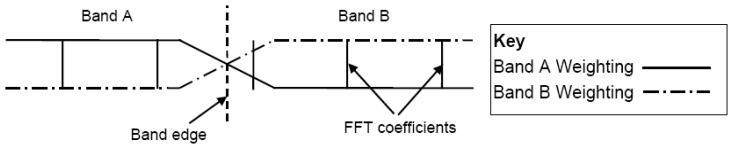


Fig. 3.25. Conversion from linear to log scale

- Audio Spectrum Centroid (*ASC*) – describes the center of gravity of the log-frequency power spectrum. Power spectrum coefficients below 62.5 Hz are replaced by a single coefficient, with the power equal to their sum and a nominal frequency of 31.25 Hz. Frequencies of all coefficients are scaled to an octave scale anchored at 1 kHz. The spectrum centroid is calculated as follows:

$$C = \sum_n \log_2\left(\frac{f(n)}{1000}\right) \cdot P_x(n) / \sum_n P_x(n) \tag{3.87}$$

where $P_x(n)$ is the power associated with frequency $f(n)$. The mean value and the variance of the spectrum centroid over time are denoted as *ASC* and *ASCv*, respectively.

- Audio Spectrum Spread (*ASS*) – describes the spread of the log-frequency power spectrum (the second moment of the log-frequency

power spectrum). To be coherent with other descriptors, in particular with ASE, the spectrum spread is defined as the RMS deviation of the log-frequency power spectrum with respect to its center of gravity:

$$S = \sqrt{\frac{\sum_n \left(\log_2 \left(\frac{f(n)}{1000} - C \right)^2 \cdot P_x(n) \right)}{\sum_n P_x(n)}} \tag{3.88}$$

where C is the spectrum centroid. The mean value and the variance of S over time are denoted as *ASS* and *ASSv*, respectively.

- Audio Spectrum Flatness (*SFM*) – describes the properties of the short-term power spectrum of an audio signal. This descriptor expresses deviation of the signal power spectrum from a flat spectral shape for a given band. The spectral flatness analysis is calculated for a number of frequency bands between 250 Hz and 16 kHz. A logarithmic frequency resolution of ¼ octave is used for all bands. This gives a total number of 24 bands in every spectral frame. For each frequency band, the spectrum flatness measure is defined as the ratio of the geometric and the arithmetic mean of the power spectrum coefficients $c(i)$ within the band b (i.e. from coefficient index il to coefficient index ih, inclusive):

$$SFM_b = \frac{\left(\prod_{i=il(b)}^{ih(b)} c(i) \right)^{\frac{1}{ih(b)-il(b)+1}}}{\frac{1}{(ih(b)-il(b)+1)} \sum_{i=il(b)}^{ih(b)} c(i)} \tag{3.89}$$

The mean values and variances of each SFM_b over time are denoted as $SFM_1 \ldots SFM_{24}$ and $SFMv_1 \ldots SFMv_{24}$, respectively.

Timbre Descriptors

- Log-Attack-Time (*LAT*) – is defined as the logarithm (decimal basis) of time duration between the time the signal starts (T_0) and the time it reaches its sustained part (T_1):

$$LAT = \log_{10}(T_1 - T_0) \tag{3.90}$$

- Temporal Centroid (*TC*) – is defined as the time averaged over the energy envelope *SE*:

$$TC = \frac{\sum_{n=1}^{length(SE)} n/sr \cdot SE(n)}{\sum_{n=1}^{length(SE)} SE(n)} \tag{3.91}$$

where *sr* is the sampling rate.

- Spectral Centroid (*SC*) – is computed as the power weighted average of the frequency of bins in the power spectrum (*Instantaneous Spectral Centroid - ISC*):

$$ISC = \frac{\sum_{k=1}^{length(S)} f(k) \cdot S(k)}{\sum_{k=1}^{length(S)} S(k)} \tag{3.92}$$

where $S(k)$ is the kth power spectrum coefficient and $f(k)$ stands for the frequency of the kth power spectrum coefficient. The mean value and the variance of a spectrum centroid over time are denoted as *SC* and *SCv*, respectively.

- Harmonic Spectral Centroid (*HSC*) – is the average of the *Instantaneous Harmonic Spectral Centroid* (*IHSC*) values computed in each frame. They are defined as the amplitude (linear scale) weighted mean of the harmonic peaks of the spectrum:

$$IHSC = \frac{\sum_{h=1}^{nb\_h} f(h) \cdot A(h)}{\sum_{h=1}^{nb\_h} A(h)} \tag{3.93}$$

where *nb_h* is the number of harmonics taken into account, $A(h)$ is the amplitude of the harmonic peak number h and $f(h)$ is the frequency of the harmonic peak number h. The mean value and the variance of a harmonic spectrum centroid over time are denoted as *HSC* and *HSCv*, respectively.

- Harmonic Spectral Deviation (*HSD*) – is the average of the *Instantaneous Harmonic Spectral Deviation* (*IHSD*) values computed in each frame. They are defined as the spectral deviation of log-amplitude components from the global spectral envelope:

$$IHSD = \frac{\sum_{h=1}^{nb\_h} \left| \log_{10} A(h) - \log_{10} SE(h) \right|}{\sum_{h=1}^{nb\_h} \log_{10} A(h)} \tag{3.94}$$

where $SE(h)$ is the local spectrum envelope (mean value of the three adjacent harmonic peaks) around the harmonic peak number h. To evaluate the ends of the envelope (for h=1 and h=nb_h) the mean amplitude of two adjacent harmonic peaks is used. The mean value and the variance of harmonic spectrum deviation over time are denoted as *HSD* and *HSDv*, respectively.

- Harmonic Spectral Spread (*HSS*) – is computed as the amplitude weighted standard deviation of the harmonic peaks of the spectrum, normalized by the harmonic spectral centroid:

$$IHSS = \frac{1}{IHSC} \sqrt{\frac{\sum_{h=1}^{nb\_h} A^2(h) \cdot \left(f(h) - IHSC \right)^2}{\sum_{h=1}^{nb\_h} A^2(h)}} \tag{3.95}$$

where *IHSS* is the Instantaneous Harmonic Spectral Spread, *IHSC* is the harmonic spectrum centroid. The mean value and variance of the harmonic spectrum spread over time are denoted as *HSS* and *HSSv*, respectively.

- Harmonic Spectral Variation (*HSV*) – is defined as the normalized correlation between the amplitude of the harmonic peaks of two adjacent frames:

$$IHSV = 1 - \frac{\sum_{h=1}^{nb\_h} A_{-1}(h) \cdot A(h)}{\sqrt{\sum_{h=1}^{nb\_h} A_{-1}{}^2(h)} \cdot \sqrt{\sum_{h=1}^{nb\_h} A^2(h)}} \tag{3.96}$$

where *IHSV* is the Instantaneous Harmonic Spectral Variation, $A(h)$ is the amplitude of the harmonic peak number h at the current frame and $A_{-1}(h)$ is the amplitude of the harmonic peak number h at the preceding frame. The mean value and the variance of harmonic spectrum variation over time are denoted as *HSV* and *HSVv*, respectively.

Peeters et al. (2000) used parameters from Timbral Spectral and Timbral Temporal groups for experiments on perceptual features of harmonic and percussive sounds. They create five-dimensional space (see Fig. 3.26) while describing instruments characterized by harmonic spectrum (*Harmonic Timbre Space*), and three-dimensional space (see Fig. 3.27) for percussive instruments (*Percussive Timbre Space*).

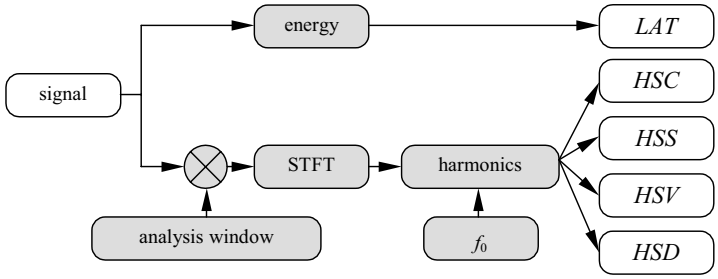


Fig. 3.26. Parameter extraction of instruments with harmonic spectrum (http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm)

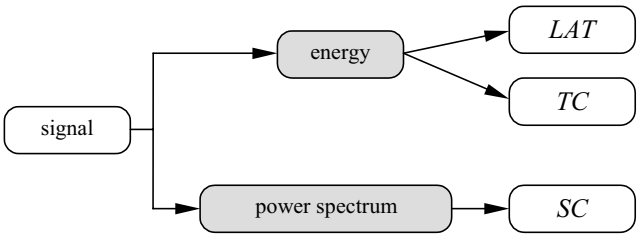


Fig. 3.27. Parameter extraction of percussive instruments (http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm)

Summarizing the information on musical instruments presented in this chapter, it may be said that there is not yet consensus on the choice of parameters for musical instrument sound description, even if some parameters are already standardized within the MPEG-7 framework. Consequently, sound feature extraction is in principle a multi-dimensional process that should be optimized based on some experimental procedures customized for each individual application field.

3.3 Artificial Neural Networks

Neural networks have proved to be important tools in decision making over a broad spectrum of applications, including such tasks as classifica-

tion and cluster analysis of data. Systems based on these algorithms have become especially significant in the processes of speech and image recognition, or optical character recognition. Applications of musical sound classification have also appeared (Christensen et al 1992; Czyzewski 1997; Kostek 1994; Kostek and Krolikowski 1997; Morando 1996; Mourjopoulos and Tsoukalas, 1991). The latter usage has become one of the most interesting areas within the broader field of musical acoustics.

In the course of rapid development of artificial neural networks numerous neuron models were proposed (Cimikowski and Shope 1996; Knecht et al 1995). The most common neural network model is the multilayer perceptron (MLP). Neural structures used nowadays are usually based on the enhanced McCulloch-Pitts model (Zurada 1992; Zurada and Malinowski 1994) which involves modifying the neuron activation function (NAF). Although continuous sigmoidal functions are still most widely used as NAFs, radial basis functions are now being encountered with increasing regularity; they are used in radial basis function (RBF) and in hyper-radial basis function (HRBF) networks. Both types of artificial neural networks are applied to problems of supervised learning (e.g. regression, classification and time series prediction). In recent years, a variety of artificial neural network classifiers were developed. Much attention was paid both to network architectures and learning algorithms. Today a large collection of neural algorithms is available, which can be used in modeling dependencies, processes and functions. Besides NN basic topology such as perceptron, Hopfield networks, bidirectional associative memory (BAM) networks or their transformations are also at our disposal.

Individual neuron units are interconnected, forming a neural network. Generally, it can have an almost arbitrary structure, however, certain limitations remain valid, as effective algorithms for teaching such irregular networks have not yet been engineered. For this reason, neural structures are regular, and depending on the structure, they can be classified as feedforward networks, recurrent networks or cellular networks. Moreover, regular networks can be combined into larger structures called modular networks, depending on the mapping which they perform. An original architecture was proposed by Fukushima along with an appropriate training method (Fukushima 1975, 1988; Fukushima and Wake 1991). The *cognitron* structure and subsequently the *neocognitron* structure are modeled after the human visual nervous system; they are designed for robust visual pattern recognition. Neocognitron is a self-organized, competitive learning, hierarchical multilayer network. It is useful for pattern classification without supervised learning, especially when there are possible shifts in the position or the distortion of a shape.

A difficult and yet very important problem of neural network design is the selection of network topology, i.e. the number of neurons in the individual layers of networks. The number of neurons must not be too small, since the network would be unable to map all possible states of solution cases. On the other hand, the network cannot be too large, as this would cause the loss of its capability to generalize, i.e. the ablility to generalize about unknown cases from the acquired knowledge to the benefit of memory learning. Usually, after a long trial-and-error process, an oversized topology is chosen which is prone to such drawbacks as a high demand on computational resources and a high generalization error. A way to solve this is to use so-called pruning methods (Karnin 1990). Two methods for optimizing network size may be used:

- minimizing the cost function (weights of the smallest influence on a cost function can be removed);
- penalty function is imposed on ineffective (unnecessary) neural structures to find the simplest solution.

In both cases the algorithm causes either the weight or the whole neuron to be ignored. The first solution seems more precise, however it is very time-consuming and therefore highly inefficient as far as network training time is concerned. The method utilizing a penalty function is simple and still relatively effective. Examples of pruning algorithms will be shown further on.

Artificial Neural Networks have the ability of learning and adapting to new situations by recognizing patterns in previous data. A neural network processes an input object by using the knowledge acquired during the training phase. The methods of training are often divided into two basic classes: training with a teacher (with supervision) and without a teacher (without supervision).

In the case of supervised learning, pattern-class information is used. It requires a desired output in order to learn. The goal of this type of network is to create a model that correctly maps the input to the output using historical data, so that the model can then be used to produce the proper output when it is unknown. An unknown probability density function $p(x)$ describes the continuous distribution of patterns x in the pattern space R^n. During the process of learning, an accurate estimation of $p(x)$ is searched for. Supervised learning algorithms depend on the class membership of each training sample x. Class-membership information allows the detection of pattern misclassifications and the computation of an error signal. The error information then reinforces the learning process. Unsupervised learning systems use unlabelled pattern samples. They adaptively gather pat-

terns into clusters or decision classes D_j. In the case of neural networks, supervised learning is understood as a process in which the gradient descent in the space of all possible synaptic-values is estimated. In the subject bibliography, two main unsupervised learning methods are covered: the Hebb method and the competition method. Unsupervised learning is a data mining technique used in clustering, data compression and principal component analysis (PCA).

Artificial Neural Networks, in general, may be classified as feedforward and feedback types depending on the interconnection type of neurons. At present, multilayer networks of the feedforward type, which are trained using the error backpropagation method (*EBP*), are applied to the majority of applications employing neural computing (Bershad et al 1993a, 1993b; Magoulas et al 1997; Werbos 1988).

Multilayered feedforward networks have, however, some essential drawbacks. Among these are the possibility of poor training convergence, difficulties in setting optimal or suboptimal values of learning parameters which then influence the convergence, the feasibility of being trapped in local minima, and poor generalization in the case of improper network size. The first three problems can be partially solved by assigning variables as learning parameters. The variables could change according to the convergence rate and training development. On the other hand, the problem related to the neural network topology is generally still unsolved. However, as mentioned before, there are some techniques, called weight pruning algorithms, that enable better network design (Karnin 1990). The basic principles of such algorithms will be further examined.

Since Artificial Neural Networks (ANN) have become standard tools in many domains, only the main features of such algorithms will be reviewed in this chapter, especially those which were adopted in the experiments.

3.3.1 Neural Network Design

Computational power of neural networks is not derived from the capabilities of a single neuron, but it rather comes from the immense structure of their interconnections. Considering structure, neural networks can be divided into three groups:

- feedforward networks,
- recurrent networks,
- self-organizing networks.

The design and operation of a feedforward network is based on a net of artificial neurons. The simplest case of a neural network is a single neuron.

The artificial neuron consists of a processing element, input signals $\boldsymbol{x}=[x_1,x_2,x_3,...,x_N]^T \in R^N$ and a single output o (Fig. 3.28). The output vector is defined as (Zurada 1992):

$$\boldsymbol{o} = f(\boldsymbol{w}^T\boldsymbol{x}-w_0) \tag{3.97}$$

where $\boldsymbol{w}$ is the synaptic weight vector:

$$\boldsymbol{w}=[w_1,w_2,w_3,...,w_N]^T \tag{3.98}$$

w_0 is the threshold of the neuron and f is a neuron activation function.

As may be seen in Fig. 3.28, each of the input signals flows through a synaptic weight. The summing node accumulates all input-weighted signals and then passes them to the output through the transfer function (f). The commonly used activation functions are of sigmoidal type (unipolar, bipolar, threshold, hyperbolic tangent, etc.; see examples in Fig. 3.29). The sigmoidal transfer function is given by the following formula:

$$f(\cdot) = \frac{1}{1+\exp(-\alpha \cdot x)} \tag{3.99}$$

where α is the coefficient or gain which adjusts the slope of the function that changes between the two asymptotic values (0 and +1). This function is nonlinear, monotonic and differentiable. Since the error back-propagation method using the *delta* learning rule requires a differentiable function, the sigmoidal transfer function is of special interest in most applications.

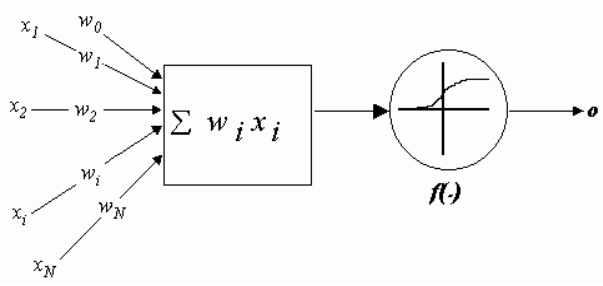


Fig. 3.28. Artificial neuron model

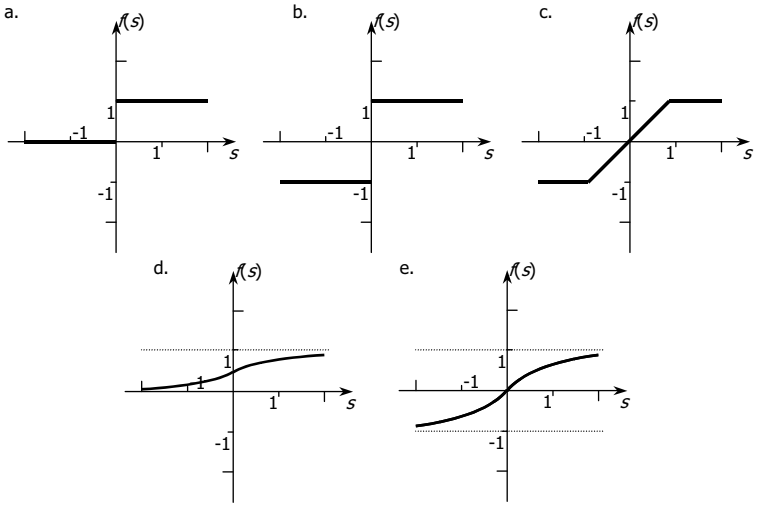


Fig. 3.29. Examples of neural network activation functions: unipolar binary (a), bipolar binary (b), bipolar threshold linear (c), sigmoid (d), hyperbolic tangent (e)

A two-layer network of the feedforward type is one of the most commonly used structures (see Fig. 3.30).

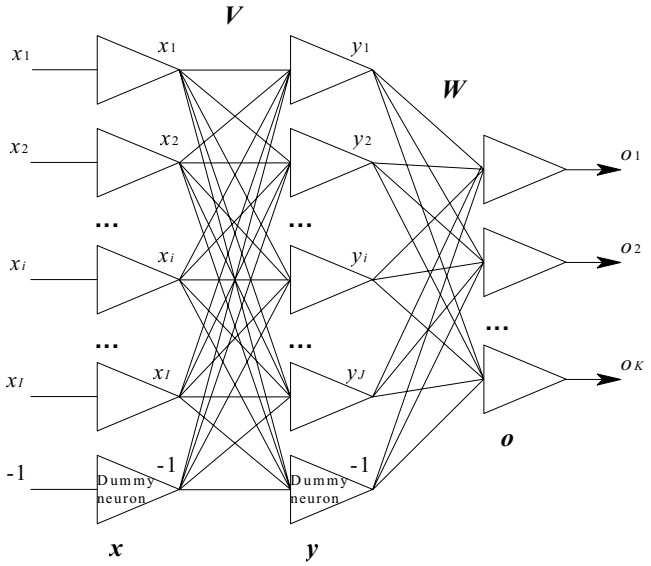


Fig. 3.30. Feedforward multi-layer network

The vector and matrix notation is more convenient for dealing with inputs, weights and outputs. The consecutive layers are denoted as input layer $\boldsymbol{x}$, hidden layer $\boldsymbol{y}$ and output layer $\boldsymbol{o}$. The number of neurons for the consecutive layers is $\boldsymbol{x}$ - I, $\boldsymbol{y}$ - J, and $\boldsymbol{o}$ - K, respectively. Let $\boldsymbol{V}(J+1 \times I+1)$ and $\boldsymbol{W}(K \times J+1)$ be, respectively, the input-to-hidden layer and the hidden-to-output layer synaptic weights. The input and hidden layers may have an additional dummy neuron each. The output value of the neuron is constant and equals -1, whereas the value of the weight may change. The dummy neuron is therefore an equivalent of the threshold synapse for all neurons in the next layer (see Fig. 3.30).

3.3.2 Recurrent Networks

A characteristic feature of recurrent networks is the feedback from network output to network input (Fig. 3.31). This means that the output signal depends not only on current input signals, but also on the whole history of excitations. A recurrent network is therefore a dynamic network.

Such networks are employed for time-consuming and computationally complex optimization processes, especially linear programming problems, i.e. the minimization or the maximization of a function value within the limits imposed on its arguments. Feedback networks are formed when the output of at least one neuron is connected directly or indirectly to its input (Fig. 3.31). Extensive bibliography describes various topologies of recurrent networks, among them is an interesting design proposed by Elman (1990). Fig. 3.31 presents a network of this type, in which output signals from the hidden layer are delayed by one z-1 cycle and then fed onto its input. Recurrent networks are the generalization of feedforward networks and are successfully employed for processing time sequences. The most often used recurrent structure is a discrete Hopfield network, composed of a single layer of neurons. Hopfield networks, for which the activation function is the signum function, are a class of networks of interesting parameters. In a Hopfield network, the output signal v_n^{k+1} from nth neuron in moment k+1 is fed onto the network input with a unitary delay.

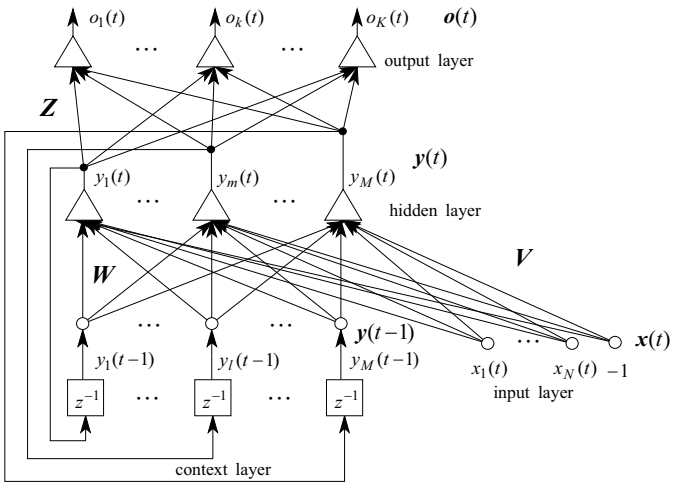


Fig. 3.31. Examples of feedback networks: generic recurrent network

3.3.3 Cellular Networks

The topology of cellular networks is based on any regular geometric structure, usually on a flat rectangular grid. A network built on it is composed of neurons forming I rows and J columns. Arbitrary cell c_{ij} located in ith row and jth column is directly connected only to neurons within a neighborhood radius, marked in Fig. 3.32 with a dashed line.

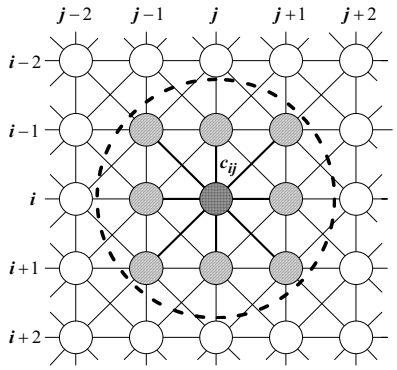


Fig. 3.32. Cellular network structure

3.3.4 Gradient-Based Methods of Training Feedforward Networks

Feedforward neural networks are subject to intensive research due to their practical use. Their concept originates from adaptive filtration and their training methods come from well-known methods of estimating weight factors of filters. A separate group of methods are modifications of the Widrow-Hoff LMS method using ADALINE units. Recently training algorithms based on the recurrent least-square (RLS) method or directly related to Kalman filtration were proposed.

The training problem may also be approached from another direction. Neural network training is equivalent to a certain optimization task and therefore one can attempt to solve it by adapting optimization theory tools, among which gradient-based methods constitute the most important class. These methods are most effective in the case of feedforward networks and that is why the present section presents the most important gradient-based methods.

Heuristic Algorithms

Gradient-based optimization methods presented so far are either excessively demanding computationally or memory-wise (Newtonian methods) or are slow-converging (the delta rule). A separate group of algorithms combining the efficiency of Newtonian methods with low computational complexity of the greatest-gradient method is constituted by heuristic algorithms. Some selected methods are presented in the following part of the Chapter.

Quickprop Algorithm

The Quickprop algorithm, proposed by Fahlman (Fahlman 1988, 1991; Fahlman and Lebiere 1991), uses information about the curvature of the mean squared error (MSE) surface. This requires the computation of the second order derivatives of the error function. The algorithm assumes the error surface to be locally quadratic and attempts to predict the behavior of the curve as it descends. Quickprop accelerates the backpropagation/gradient descent learning rate by calculating the current slope of the parabolic curve (MSE versus weight value). The descent equation takes into consideration the past and the current slope, as well as the past difference in the weight value, to calculate the next weight step which is then applied to each weight value w_{ij} separately. The derivatives are computed

in the direction of each weight. Assuming that the error function term S_{ij}^k is described by the following expression (Fahlman 1988, 1991; Fahlman and Lebiere 1991):

$$S_{ij}^k = \nabla E\left(\Delta w_{ij}^k\right) + \gamma \cdot \Delta w_{ij}^k,\ \gamma = 10^{-4}, \tag{3.100}$$

on the basis of the direction of the weight change w_{ij} in step k, S_{ij}^k, and in step (k- 1), S_{ij}^{k-1}, the parabola minimum can be determined as follows:

$$\beta_{ij}^k = \frac{S_{ij}^k}{S_{ij}^{k-1} - S_{ij}^k} \tag{3.101}$$

The Fahlman rule for the weight increment Δw_{ij}^k is as follows (Fahlman 1988, 1991; Fahlman and Lebiere 1991):

$$\Delta w_{ij}^k = -\eta^k \cdot S_{ij}^k + \alpha_{ij}^k \cdot \Delta w_{ij}^{k-1} \tag{3.102}$$

where values of the learning rate η^k and the momentum ratio α_{ij}^k change according to the following expressions:

$$\eta^k = \begin{cases} \eta_0 & ;\text{if} \quad k = 1 \quad \vee \quad S_{ij}^k \cdot \Delta w_{ij}^{k-1} > 0 \\ 0 & ;\text{if} \quad k \neq 1 \quad \wedge \quad S_{ij}^k \cdot \Delta w_{ij}^{k-1} \leq 0 \end{cases} \tag{3.103}$$

$$\alpha_{ij}^k = \begin{cases} \alpha_{\max} & ;\text{if} \quad \beta_{ij}^k > \alpha_{\max} \quad \vee \quad S_{ij}^k \cdot \Delta w_{ij}^{k-1} \cdot \beta_{ij}^k < 0 \\ \beta_{ij}^k & ;\text{if} \quad \beta_{ij}^k \leq \alpha_{\max} \quad \wedge \quad S_{ij}^k \cdot \Delta w_{ij}^{k-1} \cdot \beta_{ij}^k \geq 0 \end{cases} \tag{3.104}$$

in which constant values of training parameters are chosen empirically and they satisfy the following conditions: $0{,}01 \leq \eta_0 \leq 0.6$ and $\alpha_{\max} = 1.75$.

Rprop Algorithm

Rprop (Resilient backPROPagation) is a local adaptive learning scheme, performing supervised batch learning in multi-layer perceptrons (Riedmiller 1994; Riedmiller and Braun 1993). The basic principle of Rprop is to eliminate the influence of the partial derivative size on the weight step. Both gradient descent and Quickprop algorithm (Fahlman 1988, 1991; Fahlman and Lebiere 1991) use the derivative (slope of graph, or rate of change) of the MSE with respect to the weight value. Rprop does not consider the actual value given by the derivative, but the sign of the de-

rivative, or the direction of the curve (upward or downward slope). Rprop calculates the weight step by comparing the previous slopes with the current slope. These two portions of the algorithm allow it to adapt to unexpected behaviors in the MSE versus weight graph. As a consequence, only the sign of the derivative is considered to indicate the direction of the weight update. Similarly to the Quickprop algorithm, in the RPROP algorithm the weight update rules apply to each weight individually and are concerned only with the derivative sign $\nabla E(w_{ij}^k)$, ignoring its value. Low and high limits of weight increase, i.e. $\Delta_{\min} = 10^{-6}$ and $\Delta_{\max} = 50$ were also introduced. The principles of the RPROP algorithm can be written down as follows:

$$\Delta w_{ij}^k = -\Delta_{ij}^k \cdot \operatorname{sgn}\left(\nabla E(\Delta w_{ij}^k)\right) \tag{3.105}$$

where the weight increase, called 'update-value' Δ_{ij}^k is determined as follows:

$$\Delta_{ij}^k = \begin{cases} \min\left\{\mu^+ \cdot \Delta_{ij}^{k-1}, \Delta_{\max}\right\} & ;if \quad \nabla E\left(\Delta w_{ij}^k\right) \cdot \nabla E\left(\Delta w_{ij}^{k-1}\right) > 0 \\ \max\left\{\mu^- \cdot \Delta_{ij}^{k-1}, \Delta_{\min}\right\} & ;if \quad \nabla E\left(\Delta w_{ij}^k\right) \cdot \nabla E\left(\Delta w_{ij}^{k-1}\right) < 0 \\ \Delta_{ij}^{k-1} & ;if \quad \nabla E\left(\Delta w_{ij}^k\right) \cdot \nabla E\left(\Delta w_{ij}^{k-1}\right) = 0 \end{cases} \tag{3.106}$$

where the factors μ^+ and μ^- meet the condition: $0 < \mu^- < 1 < \mu^+$. Every time the partial derivative of the corresponding weight w_{ij} changes its sign, which indicates that the last update was too big and the algorithm has jumped over a local minimum, the update-value is decreased by the factor μ^-. If the derivative retains its sign, the update-value is slightly increased in order to accelerate convergence in shallow regions. Additionally, in the case of a change in sign, there should be no adaptation in the succeeding learning step. The original Rprop algorithm assumed $\mu^- = 0.5$ and $\mu^+ = 1.2$.

3.3.5 Application of Pruning Weight Algorithms

The common problem of all neural networks is selecting an appropriate size of the structure. Attempts to optimize the neural network structures have been made, in order to avoid overfitting and improve generalization, to obtain higher convergence speed and less costly implementations. There are several techniques of optimizing the neural architecture. Basically, they

are: empirical methods, methods based on statistical criteria, growing (constructive) methods, decreasing (destructive, pruning) methods, and hybrid methods.

The most common possibilities are sensitivity algorithms and penalty term algorithms (e.g. Backpropagation with Weight Decay), which are mentioned in this chapter. In both cases, a weight pruning algorithm results in the neglect of either weights or a neuron. The first solution (sensitivity) seems to be more robust, however the methods tested (*Optimal Brain Damage* - OBD, *Optimal Brain Surgeon* - OBS) are time-consuming and thus ineffective in terms of training duration. Magnitude Based Pruning is the simplest weight pruning algorithm. After each training, the link with the smallest weight is removed. Thus the salience of a link is just the absolute size of its weight. Though this method is very simple, it rarely yields worse results than some more sophisticated algorithms. In the Optimal Brain Damage Algorithm (OBD) method, the feedforward neural network (FANN) is trained and the weight saliencies are calculated. The weights with the lowest saliencies are eliminated and finally, the network is retrained. Optimal Brain Surgeon (OBS) is a further development of OBD. It leads to a more exact approximation of the error function. Also, a so-called skeletonization prunes units by estimating the change of the error function when the unit is removed (like OBS and OBD do for links). For each node, a so-called attentional strength is introduced which leads to a different formula for the net input (Karnin 1990).

Non-contributing units is yet another method that deals with an oversized net structure. This method uses statistical means to find units that do not contribute to the net behavior. The net is subdivided into its layers, the output of each neuron is observed for the whole pattern set. Units are removed if their output does not vary, or if they always show the same output as any other unit of the same layer, or if they show the output which is opposite to the output of any other unit in the same layer.

Penalty Functions

A separate group of methods are those relying on the modifications of the target function. They were designed for a quadratic error function $E(\mathbf{w})$ and they rely on eliminating small and least significant weight connections by adding a $C(\mathbf{w})$ term called a penalty term, a regularizer term, or a forgetting term (Ishikawa 1996, 1997). The term penalizes weight values that are too large. This leads to a gradual decrease of weights down to a threshold value, beyond which they are zeroed. The overall cost function $C_{total}(\mathbf{w})$ can be described by the formula:

$$C_{total}(\mathbf{w}) = E(\mathbf{w}) + C(\mathbf{w}) \tag{3.107}$$

then during each cycle a training of two following steps is composed of:

- minimizing the value of the target function $E(\mathbf{w})$ using one of the error back-propagation methods,
- correcting the values after adding the penalty term.

Various types of penalty functions $C(\mathbf{w})$ are proposed in literature. Among the most popular ones are those proposed by Rumelhart et al. (1986). It is worth noting that penalty functions can also be used for removing unnecessary neurons in hidden layers.

Even a simple evaluation of neuron influences requires additional training. On the other hand, methods with the penalty function are simple and quite efficient. They may also be used to obtain a skeleton network structure during the process of rule discovery.

In the case of the weight pruning algorithm, for the weight w_{ij} the cost function E is modified as follows:

$$E'(\mathbf{W}) = E(\mathbf{W}) + \frac{1}{2} \cdot \gamma \cdot \sum_{i,j} \frac{w_{ij}^2}{1 + w_{ij}^2} \tag{3.108}$$

where γ is a positive constant. The error backpropagation for the weight adjustment is therefore as follows:

$$w_{ij}' = w_{ij} \cdot \left(1 - \eta \cdot \gamma \cdot \frac{1}{\left[1 + w_{ij}^2\right]^2} \right) \tag{3.109}$$

Largely discussed during the last few years, pruning methods lead to compact networks, which show good performance as compared to the starting architecture or to other structures of greater size. Though the resulting configuration is sparsely connected, but not symmetrical. Therefore, in literature one may find studies dealing with quantitative comparison of pruning algorithms with symmetry constraints, for feedforward neural networks.

Fahlman's Cascade-Correlation Algorithm

In a series of papers Fahlman analyzed the problem of teaching multi-layer feedforward and recurrent networks, and proposed an original method of

constructing the neural network concurrently with training it (Fahlman 1988, 1991; Fahlman and Lebiere 1991). His concept is based on limiting the teaching process of certain neurons and weighted connections, so that in each training step only one hidden neuron is subject to change. The detailed analysis of the proposed optimal network construction algorithm can be found in the appropriate literature.

3.3.6 Unsupervised Training Methods

One can distinguish two types of unsupervised learning methods, namely those using competition and those based on the Hebb rule. The latter are usually applied to data compression (PCA) and blind signal separation. Competition-based methods remain most popular, as they present an attractive alternative to classic vector quantization (VQ) techniques employed for image compression and speech compression, among others. This research field greatly benefited from the works of Kohonen (1990; Kohonen et al 1996, 1997), who introduced the name of self-organizing maps (SOMs) for networks trained using competition-based methods (Kohonen 1990; Kohonen et al 1996, 1997). Although a number of alternative self-organization approaches were proposed, SOM networks remain researchers' center of attention (Amerijck et al 1998; Choi and Park 1994; Flangan 1996, 1997). Therefore this section presents the solutions proposed by Kohonen (1990; Kohonen et al 1996; Kangas et al 1990). It is also worth noting that an interesting class of networks are those whose training and functioning is based on the adaptive resonance theory (ART). Their attractiveness results from the fact that the number of categories into which input data is classified, is not known a priori and is determined dynamically in the course of the algorithm. For that reason ART networks have been the subject of numerous papers and modification (Carpenter and Markuzon 1998; Carpenter et al 1991; Frank 1998).

SOM-Type Self-Organizing Networks

According to the idea of a SOM network, each neuron becomes a certain template for a group of similar input signals because of its weight vector, while adjacent neurons represent close areas of these templates. Therefore such a network maps the input vector space into its internal structure, depending on the chosen metrics. This structure does not reflect the actual connections between neurons and is used only for determining their neighborhoods. It can have any arbitrary shape, however in literature it is usually chosen to be two-dimensional (rectangular or hexagonal).

Formally speaking, SOM networks define the mapping o *N*-dimensional input signal space into a two-dimensional regular neural structure modified by means of competition between neurons forced by the *x* input vector. This competition is won by the *c*th output unit, if the following relation is true:

$$d(\mathbf{x}, \mathbf{W}_c) = \min_{1 \le i \le K \times L} d(\mathbf{x}, \mathbf{W}_i) \text{ or } c = \arg\min_{1 \le i \le K \times L} \{d(\mathbf{x}, \mathbf{W}_i)\} \tag{3.110}$$

where d is a measure of distance between vector *x* and the weights vector W_i of an output vector in a $K \times L$ two-dimensional structure.

The process of training a SOM network can be described in the categories of minimizing the error function *E*, when the weight adaptation in the *k*th iteration is described by the following relation:

$$\forall_{i \in N_c} \mathbf{W}_i^{(k+1)} = \mathbf{W}_i^{(k)} + \Delta \mathbf{W}_i^{(k)} = \mathbf{W}_i^{(k)} - \eta^{(k)} \cdot \nabla_{\mathbf{W}_i} E^{(k)} \tag{3.111}$$

where N_c is a set of units adjacent to the winning neuron, $\eta \in [0,1]$ defines the speed of learning, while the function *E* associated with vector quantization is described by the following expression:

$$E^{(k)} = \sum_{i=1}^{K \times L} h_{ci}^{(k)} \cdot \Psi\left[d(\mathbf{x}^{(k)}, \mathbf{W}_i)\right] \tag{3.112}$$

where h_{ci} is a function of spatial neighborhood for the *c*th winning unit, while Ψ is a function of the distance measure *d*. Measure *d* is universally assumed to be based on Euclidean metric. Taking into account that weight adaptation takes place in the neighborhood of the winning neuron and defining $\Psi(d) = d^2 / 2$, the expression describing weight updates takes the following form:

$$\forall_{i \in N_c} \mathbf{W}_i^{(k+1)} = \mathbf{W}_i^{(k)} + \eta^{(k)} \cdot h_{ci}^{(k)} \cdot \left[\mathbf{x}^{(k)} - \mathbf{W}_c\right] \tag{3.113}$$

where the starting values of weight vectors W(0) are usually chosen at random within the range [-1, 1], the value of the learning speed coefficient $\eta^{(0)}$ is assumed to be 0.95 and the neighborhood function h_{ci} can be a constant function or a Gaussian one. Learning speed coefficients η and neighborhood radii are functions decreasing monotonically as training proceeds. Recommended values of training parameters as well as expressions

describing their changes can be found in Kohonen and his co-workers' papers (1996, 1997).

Since ANNs have grown to become a useful tool in many pattern recognition applications, this suggests that they may work well in the musical signal domain, even precluding other approaches to the problem of musical instrument sound classification. The application of ANNs within the musical acoustics domain will be shown in the following sections.

3.3.7 Application of Neural Networks to Musical Instrument Sound Classification

A variety of neural network types were used for the automatic classification of musical instrument sounds. The author and her team extensively tested multilayer neural networks of different configurations for musical sound classification. Examples of such tests were published in 90s, and continued through recent years, of which some results will be shown later on (Kostek 1994; Kostek 1999; Kostek and Czyzewski 2001b; Kostek 2004a, 2004b; Kostek et al 2004, 2005). A feedforward neural network was tested in experiments carried out by Kaminskyj and his co-workers (Kaminskyj 2000; Kaminskyj and Materka 1995). They started with experiments, which aimed at classyfying the sounds of musical instruments, such as: piano, marimba, accordion and guitar. High effectiveness of 97% was attained. Lately, a more thorough study was continued by Kaminskyj (Kaminskyj 2002), which resulted in the completion of his Ph.D. work. Lately, in the works by Kaminskyj (Kaminskyj 2002; Kaminskyj and Czaszejko 2005), and separately in the research carried out by the author's team (Szczuko et al 2004, Kostek et al 2005) and also by Eronen and Klapuri (Eronen 2001; Eronen and Klapuri 2000) the attempt was made to generalize features of the classification system. To this end, different musical sound databases were used in the phases of testing and recognition, see the review in Eronen's work (2001). Cemgil and Gürgen (1997) used three different neural network topologies, namely a multi-layer perceptron, a time-delay network, and a self-organizing network (SOM) for testing a limited number of sounds. In their study 40 sounds of 10 different musical instruments were chosen from 10 different classes of octave A3-A4 (Cemgil and Gürgen 1997). The effectiveness achieved was up to 100% for the time-delay network, less for multilayer perceptron (approx. 97%), and 94% for SOM (Cemgil and Gürgen 1997).

Self-organizing maps were extensively tested by many researchers (Feiten and Günzel 1994; Cosi et al 1994a, 1994b; Cimikowski and Shope 1996; Toiviainen et al 1998), however such algorithms were used for comparison be-

tween machine-clustering of timbral spaces and the results of similarity judgments by human subjects (Grey 1997; de Poli and Prandoni 1997; Wessel 1979). In the context of timbre similarity, SOMs were used also by other researchers (Zhang and Kuo 1999; Agostini et al 2001).

Another application of SOM to musical instrument sound classification was introduced by Zhang (Zhang and Kuo 1999). Kohonen self-organizing map was employed to select the optimum structure of the feature vectors. The classification was performed by a multilayer perceptron. The system was applied to polyphonic music having a dominant instrument. The system attained approx. 80% of correct recognition. Agostini et al. used a complex topology of neural networks in their study (Agostini et al 2001). In addition, Fragoulis et al. (Fragoulis et al 1999) applied an ARTMAP for the classification of five instruments employing ten features. The accuracy achieved was very high.

An exhaustive review of research both on features and techniques used in automatic classification of musical instrument sounds was done by Herrera and co-workers (Herrera et al 2000; Herrera et al 2003). They focus on two complementary approaches to musical sound classification, namely the perceptual approach and the taxonomic approach. As seen from this review, features that are used in a research on musical sound classification may be divided into two groups, namely those based on perceptual properties of the human auditory system, and others that are determined on the basis of physical characteristics of musical instruments. Any of these features may be regarded until now as optimum ones. The same remark refers to classification techniques, any of the developed systems by far do not identify all musical instrument sounds with 100% accuracy. This review is very valuable and comprehensive, thus a Reader interested in the research on the classification of musical instrument sounds can be referred to the paper by Herrera et al., and also to the sources contained there (Herrera et al 2000, 2003).

In the next paragraphs some experiments related to musical instrument sound classification carried out in the Multimedia Systems Department will be shown. Experiments that referred to pitch detection were mostly performed by Dziubinski (Dziubinski and Kostek 2004), whereas some classification tests based on neural networks were carried out by Dalka, Dabrowski, Dziubinski, Szczuko and Kostek (Szczuko et al 2004; Kostek et al 2005; Dziubinski et al 2005), also by Kostek and Zwan and Dziubinski (2001, 2002, 2003). Parts of these studies have been performed within M.Sc. and Ph.D. works supervised by the author.

Pitch Detection

The multistage of musical sound classification often starts with a pitch detection algorithm, which apart from the detection of the fundamental frequency of a sound, identifies all other harmonic frequencies, thus enabling the calculation of many features.

In general, the evaluation of pitch detectors performance can be analyzed objectively (Rabiner et al 1976; McGogenal et al 1977) and subjectively. Pitch estimation errors are classified as *Gross Pitch Errors* and *Fine Pitch Errors,* as suggested by Rabiner et al. (1976). Gross pitch errors are here called octave errors, since the calculated pitch differs by around one (or more) octaves from the expected pitch. Objective tests of the proposed PDA, in terms of octave errors, are presented later on. Objective tests for *Fine Pitch Errors* are based on synthetic, quasi-periodic signals. Synthesized signals have time varying pitch and each harmonic amplitude varies in time independently, giving reliable simulation of real signals. In addition, comparative tests for different noise levels, contaminating test signals have been carried out. Subjective tests, showing perceptual accuracy of estimated pitch contours, were omitted in this work.

Evaluation of instantaneous pitch track (IPT) estimation based on synthetic signals

Objective tests for *Fine Pitch Errors* were performed on synthetic signals, synthesized according to the formula:

$$S[n] = \sum_{k=1}^{K} \frac{A[k][n]}{k} \cdot \sin\left(\frac{2\pi f[n]kn}{f_s} + 2\pi\varphi[k]\right), \qquad n = 1,\ldots,N \tag{3.114}$$

where:
S – synthesized signal,
A – matrix containing amplitude envelopes (for each harmonic),
K – number of harmonics,
N – number of samples,
f_s – sampling frequency,
φ - vector containing phase shifts,
f – fundamental frequency of the synthesized signal.

Vectors A[k] and f fluctuate in time, and are generated according to the following limitations:

- frequency of fluctuations is not higher than 10 Hz,
- $0.1 < A[k] < 1$
- $0.95 < f < 1$ - keeping frequency fluctuations below 5%,

- amplitudes of fluctuations (with regard to given limitations) are chosen in a pseudorandom way.

In addition, all rows of matrix **A** are generated separately (they differ from each other) and all values of phase shifts (stored in φ vector) are generated in pseudo-random way.

The proposed IPT algorithm was implemented in *Matlab* and its performance was analyzed. Pitch estimations were performed for signals with the duration of 1 second, and for a sampling frequency equal to 44100 Hz. Since instantaneous pitch of synthesized signals was known, it was possible to calculate the instantaneous error of pitch fluctuations. A set of hundred frequencies for the test signals was chosen from a frequency range of 50 Hz to 4000 Hz. In addition, for each chosen frequency, 10 signals were generated and an average error for the frequency was calculated. 1000 signals, generated in respect to Eq. 3.115, were tested.

Frequency was calculated according to the expression:

$$F[k] = f_{start} + (f_{stop} - f_{start}) \cdot \frac{k-1}{K-1} \qquad k=1,...,K \tag{3.115}$$

where:

F – set of chosen test frequencies,

K – number of chosen frequencies (K = 100),

f_{start}, f_{stop} - the lowest and the highest frequencies of the chosen set (f_{start} = 50 Hz, f_{stop} = 4000 Hz).

Pitch estimation error for each signal tested is understood to be:

$$E_{IPE} = \frac{1}{N}\sum_{n=1}^{N} 100\% \cdot \left(\frac{|IPT_e[n] - IPT[n]|}{IPT_e[n]}\right) \tag{3.116}$$

where:

E_{IPE} – average error of calculated IPT,

IPT - estimated instantaneous pitch track,

IPT_e - expected instantaneous pitch track,

N – number of samples of instantaneous pitch track (number of samples of the test signal).

The results of IPT performance tests show for all chosen frequencies that the error decreases with increasing frequency. This is due to the fact that for lower pitched sounds, harmonics are placed closer to each other. Therefore, they have greater influence on each other, firstly due to the leakage effect of the spectrum, and secondly, because a relatively smaller region of spectrum surrounding a chosen harmonic is involved in calculat-

ing instantaneous pitch. An average error for all frequencies (1000 tested signals) is equal to 0.0108 % (see Fig. 3.33).

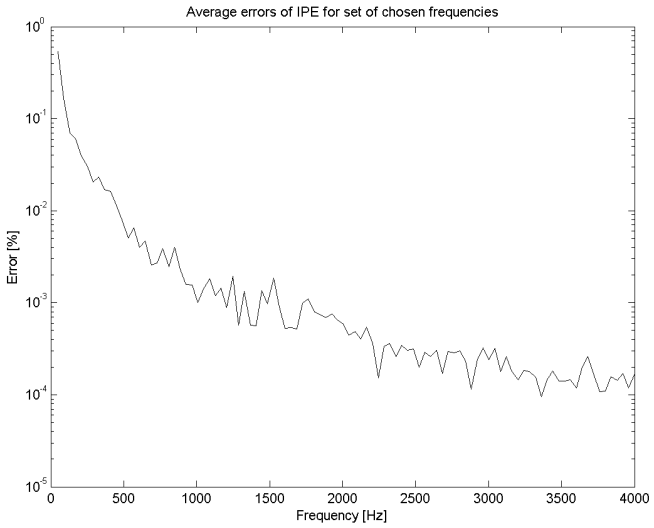


Fig. 3.33. Results of IPE performance tests for all chosen frequencies

Estimation of pitch track in noise

Because a time-domain representation of one of the input signals harmonics is calculated based on a spectrum peak related to the average pitch of the analyzed block (and its surrounding spectrum fragment), and because some spectrum bins with relatively low energy within the chosen region may be below noise level, noise disturbances are an important factor restricting performance of the IPT. Noise contaminating tested signals was of Gaussian and of the additive type. Table 3.1 presents the average estimation error for the implemented algorithm for SNRs equal to 6, 12 and 24 dB.

Table 3.1. Average IPE performance errors for different SNRs

SNR [dB]	Average error [%]
6	0.1964
12	0.1169
24	0.0549

Similarly to Fig. 3.33, Fig. 3.34 presents errors calculated for chosen frequencies, and tested SNRs.

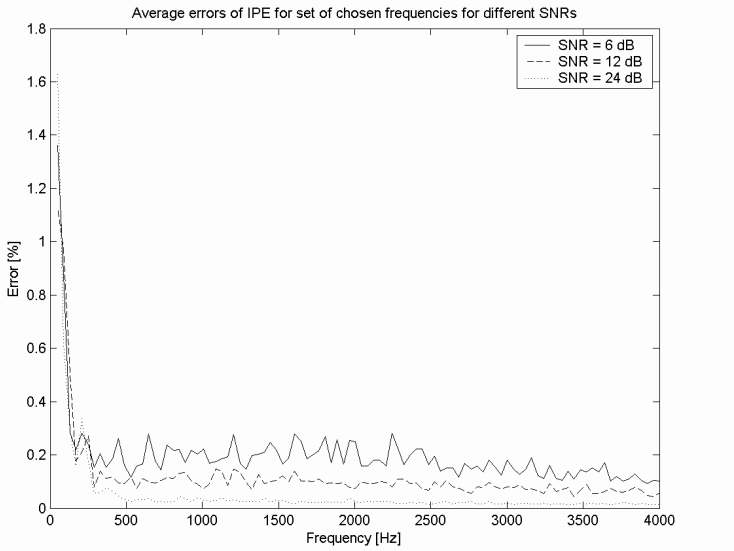


Fig. 3.34. Results of IPE performance tests for all chosen frequencies, with regard to SNR equal to 6, 12 and 24 dB

Octave errors performance

In order to determine the efficiency of a presented SPA, in terms of octave errors, 567 musical instrument sounds have been tested. Analyses of six instruments in their full scale, representing diverse instrument groups, and one instrument with all the articulation types have been performed. Sounds recorded at the Multimedia Systems Department (MSD) of the Faculty of Electronics, Telecommunications and Informatics, of Gdansk University of Technology, Poland (Kostek 1999), as well as sounds from the McGill University collection (Opolko and Wapnick 1987) were used in experiments. Tables (Tabs. 3.2-3.4) and figures (Figs. 3.35 and 3.36) present the estimated average pitch (note) played by the instrument according to the *Acoustical Society of America* (ASA) standard, and the nominal frequency of that note, specified by the ASA. In addition, pitch deviation for each estimated frequency is presented in cents, and calculated according to the formula:

$$P_{dv} = 12\log_2(\frac{\mathrm{P_{est}}}{\mathrm{P_{ASA}}}) \cdot 100\,[\mathrm{cents}] \tag{3.117}$$

where P_{dv} denotes pitch deviation of the estimated pitch (in cents) from the ASA tone frequency, P_{est} is the estimated pitch, and P_{ASA} denotes the nominal pitch of a tone as understood by ASA standard.

Results for oboe and double bass for *non legato* articulation are presented in Tables 3.2 and 3.3 Results for viola (forte, non legato) and oboe sounds played mezzo forte and with vibrato are presented in Figs. 3.35 and 3.36.

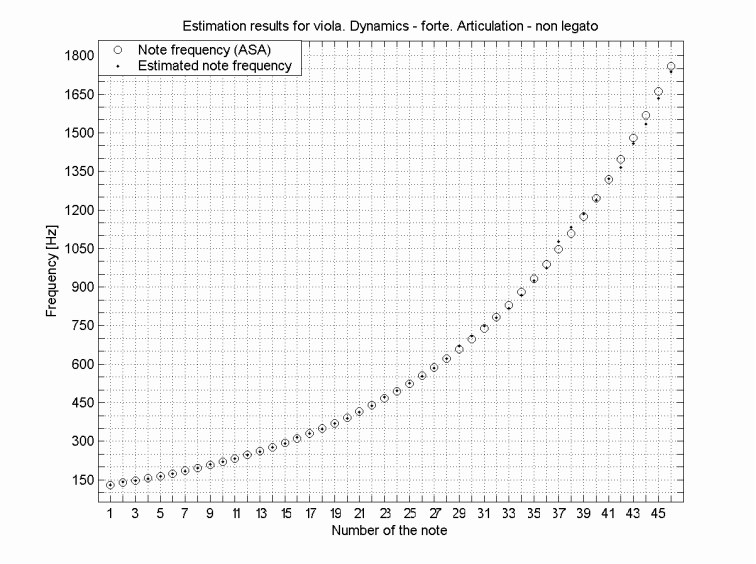


Fig. 3.35. Pitch estimation results for viola (articulation: *non legato,* dynamics: *forte*, range: C3 - A6)

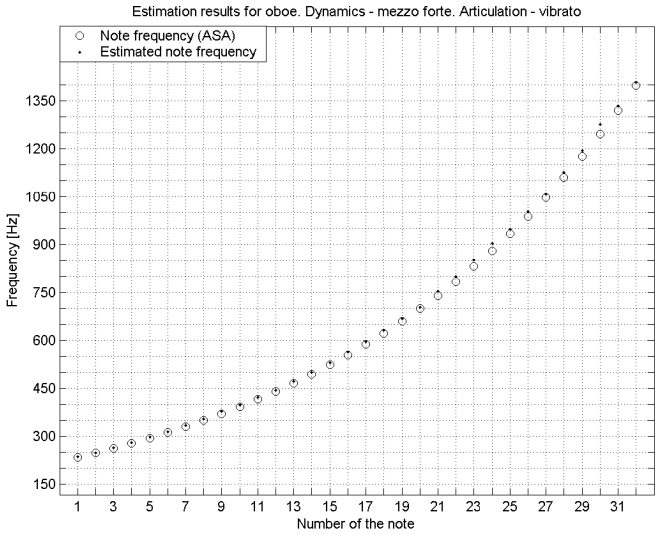


Fig. 3.36. Pitch estimation results for oboe (articulation: *non legato,* dynamics: *mezzo forte*, range: A3# - F6)

Table 3.2. Pitch estimation results for **oboe** (articulation: *non legato,* dynamics: *mezzo forte*), MSD collection

Tone (ASA)	Estimated pitch [Hz]	Nominal freq. [Hz]	Pitch deviation with regard to ASA standard [cents]	Octave error
A3#	234.24	233.08	8.6	NO
B3	245.46	246.94	-10.4	NO
C4	263.22	261.63	10.5	NO
C4#	279.8	277.18	16.3	NO
D4	295.94	293.66	13.4	NO
D4#	314.52	311.13	18.8	NO
E4	332.35	329.63	14.2	NO
F4	351.04	349.23	8.9	NO
F4#	371.95	369.99	9.1	NO
G4	394.19	392	9.6	NO
G4#	417.42	415.3	8.8	NO

Table 3.2. (cont.)

A4	442.4	440	9.4	NO
A4#	471.37	466.16	19.2	NO
B4	498.13	493.88	14.8	NO
C5	528.85	523.25	18.4	NO
C5#	563.3	554.37	27.7	NO
D5	597.98	587.33	31.1	NO
D5#	632.25	622.25	27.6	NO
E5	669.99	659.26	27.9	NO
F5	708.24	698.46	24.1	NO
F5#	755.94	739.99	36.9	NO
G5	799.07	783.99	32.9	NO
G5#	842.1	830.61	23.8	NO
A5	888.01	880	15.7	NO
A5#	936.42	932.33	7.6	NO
B5	997.3	987.77	16.6	NO
C6	1052.2	1046.5	9.4	NO
C6#	1124.5	1108.7	24.5	NO
D6	1185.5	1174.7	15.8	NO
D6#	1272.8	1244.5	38.9	NO
E6	1326.3	1318.5	10.2	NO
F6	1407.1	1396.9	12.6	NO
F6#	1502.1	1480	25.6	NO

Table 3.3. Pitch estimation results for a double bass (articulation: *non legato,* dynamics: *forte*), McGill Univ. collection

Tone (ASA)	Estimated pitch [Hz]	Nominal freq. [Hz]	Pitch deviation with regard to ASA standard [cents]	Octave error
C#1	34.54	34.648	5.4	NO
D1	36.62	36.708	4.2	NO

Table 3.3. (cont.)

D#1	38.68	38.891	9.4	NO
E1	41.1	41.203	4.3	NO
F1	43.56	43.564	0.2	NO
F1#	46.23	46.249	0.7	NO
G1	49	48.999	0	NO
G1#	51.88	51.913	1.1	NO
A1	54.97	55	0.9	NO
A1#	58.22	58.27	1.5	NO
B1	61.46	61.735	7.7	NO
C2	65.26	65.406	3.9	NO
C2#	69.12	69.296	4.4	NO
D2	73.28	73.416	3.2	NO
D2#	77.58	77.782	4.5	NO
E2	82.3	82.407	2.2	NO
F2	87.24	87.307	1.3	NO
F2#	92.23	92.499	5	NO
G2	97.63	97.999	6.5	NO
G2#	103.64	103.83	3.2	NO
A2	109.97	110	0.5	NO
A2#	116.46	116.54	1.2	NO
B2	122.99	123.47	6.7	NO
C3	130.5	130.81	4.1	NO
C3#	138.61	138.59	-0.2	NO
D3	146.3	146.83	6.3	NO
D3#	155.07	155.56	5.5	NO
E3	164.16	164.81	6.8	NO
F3	173.78	174.61	8.2	NO
F3#	184.64	185	3.4	NO
G3	195.35	196	5.8	NO

Table 3.3. (cont.)

G3#	207.36	207.65	2.4	NO
A3	219.2	220	6.3	NO
A3#	232.31	233.08	5.7	NO
B3	246.33	246.94	4.3	NO
C4	262.23	261.63	-4	NO
C4#	275.72	277.18	9.1	NO
D4	292.64	293.66	6	NO
D4#	309.59	311.13	8.6	NO
E4	329.55	329.63	0.4	NO

No octave-related errors were detected among the 567 instrument sounds processed. The engineered algorithm shows high performance for a wide variety of sounds with different fundamental frequencies, starting from 45.6 Hz for a tuba F up to 1737.6 for a viola. Since different instrument groups were analyzed and sounds played with differentiated dynamics and articulations, the proposed PDA had to deal with a large variety of situations in terms of relations between the energy of different harmonics, showing immunity to octave related errors.

Differences, sometimes significant in terms of *fine pitch errors*, between the estimated pitch and the tone frequency of a sound are caused by musicians playing solo. It happens, if the instruments are not tuned to exactly the same pitch before recording.

Automatic Pitch Detection

The pitch detection algorithm used in the automatic classification of musical instruments consists of three main stages, each of them divided into steps:

- Signal spectrum acquisition
  - Selecting a frame from the steady state of the audio signal
  - Fast Fourier Transform operation
  - Low pass filtering
  - Calculating logarithm of the spectrum amplitude
  - Trend elimination
- Harmonic peak detection
  - 1-bit quantization of the amplitude spectrum based on the assumed threshold $P1$

- Derivative calculation
- Determining harmonic peak positions

▪ Pitch detection
- Calculating differences in the positions of all detected harmonic peaks and sorting them in the ascending order
- Eliminating differences smaller than the assumed threshold $P2$
- Locating the position index of the first change of the difference values greater then the assumed threshold $P2$
- Calculating a mean value of differences having position indices smaller than the position index found in the previous step
- Expressing the mean value in Hz units

The estimated fundamental frequency f_0' is considered correct if the condition defined below is fulfilled (i.e. the difference from the real fundamental frequency f_0 is less than a semitone):

$$f_0 \sqrt[12]{2^{-1}} < f_0' < f_0 \sqrt[12]{2} \tag{3.118}$$

If Eq. (3.118) is not fulfilled, then the estimated frequency is considered to be incorrect. Detailed results of the pitch detection effectiveness are shown in Table 3.4.

Table 3.4. Effectiveness of fundamental frequency detection

Instrument	No. of sounds	No. of errors	Effectiveness [%]
bassoon	376	4	98.9
B flat clarinet	386	13	96.6
oboe	328	0	100.0
tenor trombone	358	3	99.2
French horn	334	9	97.3
alto saxophone	256	6	97.7
violin	442	56	87.3
trumpet	302	13	95.7
F flat tuba	316	20	93.7
cello	454	33	92.7
Total	3552	157	95.6

The performed experiments were based on the set of 10 instruments: bassoon, B flat clarinet, oboe, tenor trombone, French horn, alto saxophone, violin, trumpet, F flat tuba and cello. Sound samples of the instruments originated from two sources. The majority of samples (80%) came from the Catalogue of Musical Instrument Sounds, which was created in the Multimedia Systems Department of Gdansk University of Technology (Kostek 1999). The set of musical instrument sounds was complemented by the McGill University Master Samples (MUMS), giving a total number of 3500 sound samples. All samples were recorded with a sampling rate of 44.1 kHz (Opolko and Wapnick 1987).

The pitch detection algorithm had correctly estimated the fundamental frequency of 3395 audio samples out of the total of over 3500. All fundamental frequencies of the oboe sounds were detected correctly. In contrast, the worst results were obtained for a string group instruments, mainly because of the poor results for the sounds originated from MUMS (74 and 81% for the violin and cello, accordingly). The total effectiveness of almost 97% is considered as sufficient in terms of the pitch detection. All the remaining experiments regarding sound classification were based on the audio samples with the fundamental frequency estimated correctly.

Parameters

The following descriptors were taken for further analysis:

- Audio Spectrum Envelope (*ASE*)

  This results in 34 coefficients for each spectral frame. The mean values and variances of each coefficient over time are denoted as $ASE_1 \ldots ASE_{34}$ and $ASEv_1 \ldots ASEv_{34}$, respectively.
- Audio Spectrum Centroid (*ASC*)

  The mean value and the variance of a spectrum centroid over time are denoted as *ASC* and *ASCv* respectively.
- Audio Spectrum Spread (*ASS*)

  The mean value and the variance of S over time are denoted as *ASS* and *ASSv* respectively.
- Audio Spectrum Flatness (*SFM*)

  The mean values and variances of each *SFMb* over time are denoted as $SFM_1 \ldots SFM_{24}$ and $SFMv_1 \ldots SFMv_{24}$ respectively.
- Log Attack Time (*LAT*)
- Temporal Centroid (*TC*)
- Spectral Centroid (*SC*)

  The mean value and the variance of a spectrum centroid over time are denoted as *SC* and *SCv* respectively.
- Harmonic Spectral Centroid (*HSC*)

The mean value and the variance of a harmonic spectrum centroid over time are denoted as *HSC* and *HSCv* respectively.

- Harmonic Spectral Deviation (*HSD*)

The mean value and the variance of harmonic spectrum deviation over time are denoted as *HSD* and *HSDv* respectively.

- Harmonic Spectral Spread (*HSS*)

The mean value and the variance of harmonic spectrum spread over time are denoted as *HSS* and *HSSv* respectively.

- Harmonic Spectral Variation (*HSV*)

The mean value and the variance of harmonic spectrum variation over time are denoted as *HSV* and *HSVv* respectively.

- Pitch (*P*) – expresses pitch of a sound according to MIDI standard
- Content of even harmonics in spectrum (h_{ev})

Sound Descriptor Analysis

A number of parameters describing a musical instrument sound should be as low as possible because of the limited resources of computer systems. The process of decreasing the length of the feature vector is removing redundancy from the set describing an audio signal. Therefore, the evaluation criteria of the effectiveness of particular parameters have to be used for the sound classification.

Fisher statistic is often used as such a criterion. It is defined for parameter A and two classes of instruments X and Y (Kostek 1999):

$$V = \frac{\overline{A}_X - \overline{A}_Y}{\sqrt{S_{AX}^2 / n_X + S_{AY}^2 / n_Y}} \tag{3.119}$$

where $\overline{A}_X$ and $\overline{A}_Y$ are mean values of parameter A for instruments X and Y; n_X, n_Y are the cardinalities of two sets of sound parameters; and S_{AX}^2 and S_{AY}^2 are variance estimators:

$$S_{AX}^2 = \frac{1}{n_X - 1} \cdot \sum_{i=1}^{n_X} (A_{Xi} - \overline{A}_X)^2 \tag{3.120}$$

$$S_{AY}^2 = \frac{1}{n_Y - 1} \cdot \sum_{i=1}^{n_Y} (A_{Yi} - \overline{A}_Y)^2 \tag{3.121}$$

The bigger the absolute values $|V|$ of Fisher statistics are, the easier it is to divide a multidimensional parameter domain into areas representing different classes. It is much easier to differentiate between two musical instruments based on a given parameter, if its mean values for both instruments are clearly different, its variances are small and the quantity of audio samples is large.

Values of Fisher statistics calculated for selected parameters and for a selected pair of instruments are shown in Table 3.5. It was found, that for example, the *HSD* and *HSS* parameters are useful for the separation of musical sounds of different groups (brass, woodwinds, strings). Figure 3.37 shows an example of the distribution of values of these parameters obtained for instruments of similar musical scales.

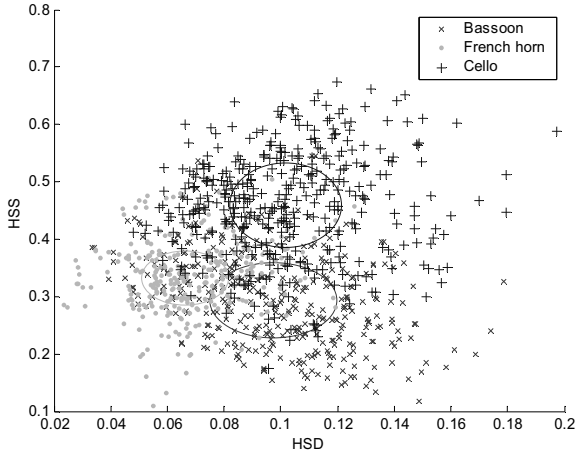


Fig. 3.37. Example of two parameter values distribution for three instruments

High value of Fisher statistic of h_{ev} parameter for the pair bassoon-clarinet proves its usefulness for the separation of clarinet sounds from other musical instruments. Initial experiments showed that Timbre Descriptors were insufficient for the effective separation of musical instruments from the same group. Therefore, the feature vector needs to be complemented by a more accurate description of a musical signal.

Better sound description is provided by the parameters connected directly with the density of the power spectrum, such as Audio Spectrum Descriptors, particularly Audio Spectrum Envelope (*ASE*) and Audio Spectrum Flatness (*SFM*). The most distinctive properties of *ASE* and *ASEv* descriptors have been noticed for the low- and high-frequency bands; mid-frequency bands are less useful for musical instrument classifi-

cation. On the other hand, *SFM* descriptors are the most accurate in mid-frequency bands. *SFMv* descriptors proved to be redundant, thus none of them have been included in the feature vector.

Table 3.5. Analysis of parameters based on Fisher statistics

pairs of instruments	h_{ev}	*LAT*	*SC*	*HSC*	*HSD*	*HSS*	*HSV*	ASE_1	ASE_2	…	SFM_{24}
bassoon - clarinet	28.79	3.22	15.73	21.25	15.12	19.41	10.72	9.10	10.56	…	14.18
bassoon - oboe	3.02	0.13	53.78	43.09	3.35	5.02	1.36	8.31	8.75	…	17.71
bassoon - trombone	1.22	0.34	12.78	11.23	19.21	17.21	0.33	8.42	9.54	…	5.46
bassoon - French horn	1.94	5.43	4.48	4.17	17.67	7.28	0.58	6.90	7.26	…	1.85
…	…	…	…	…	…	…	…	…	…	…	…
cello - tuba	4.31	15.82	12.51	15.82	26.53	16.16	5.22	4.72	0.72		22.55

A parameter similarity criterion can also be used to find dependencies between parameters. For this purpose Pearson's correlation coefficient *r* can be calculated according to the formula (Kostek 1999):

$$r = \frac{\sum_{i=1}^{n}(A_i - \bar{A})(B_i - \bar{B})}{\sqrt{\sum_{i=1}^{n}(A_i - \bar{A})^2 \sum_{i=1}^{n}(B_i - \bar{B})^2}} \tag{3.122}$$

where A_i and B_i are parameter values for a selected instrument. Absolute values of the coefficient *r* close to 1 indicate strong correlation of two parameters for the selected instrument. Table 3.6 shows calculated correlation coefficients for a bassoon. There is strong correlation between parameters related to brightness (e.g. *SC* – *HSC*, *SC* – *ASC*). Also, *ASE* coefficients for higher frequency bands show strong correlation (e.g. ASE_{24} – ASE_{28}). However, parameter dependencies express the individual character of an instrument and they differ for various instruments.

Table 3.6. Correlation coefficients *r* calculated for a bassoon

de-script.	h_{ev}	*LAT*	*SC*	*HSC*	*HSD*	…	*ASC*	…	ASE_{24}	ASE_{25}	..	ASE_{27}	ASE_{28}
Ev	1.00					…		…			..		
LAT	0.02	1.00				…		…			..		
SC	0.04	0.13	1.00			…		…			..		
HSC	0.02	0.17	0.78	1.00		…		…			..		
HSD	0.10	0.20	0.01	0.08	1.00	…		…			..		
…	…	…	…	…	…	…	…	…	…	…	..		…
ASC	0.05	0.10	0.81	0.50	0.25	…	1.00	…			..		
…	…	…	…	…	…	…	…	…	…	…	..	…	…
ASE_{24}	0.01	0.05	0.64	0.77	0.06	…	0.44	…	1.00		..		
ASE_{25}	0.05	0.16	0.64	0.84	0.16	…	0.43	…	0.79	1.00	..		
ASE_{26}	0.01	0.10	0.65	0.84	0.13	…	0.45	…	0.81	0.90	..		
ASE_{27}	0.03	0.09	0.62	0.81	0.13	…	0.42	…	0.81	0.86	..	1.00	
ASE_{28}	0.02	0.13	0.58	0.79	0.10	…	0.35	…	0.75	0.84	..	0.88	1.00

After a thorough analysis, a final content of the feature vector used for the classification purpose is as follows:

[ASE_2,…, ASE_5, ASE_8, ASE_9, ASE_{18}, ASE_{21}, ASE_{23},…, ASE_{31}, ASE_{33}, ASE_{34}, $ASEv_5$,…, $ASEv_9$, $ASEv_{21}$, $ASEv_{31}$, $ASEv_{34}$, *ASC*, *ASS*, *ASSv*, SFM_{13},…, SFM_{19}, SFM_{21}, SFM_{22}, SFM_{24}, *HSC*, *HSD*, *HSDv*, *HSS*, *HSSv*, *KeyNum*, h_{ev}, *LAT*]

Classification Results

A three-layer neural network of the feed-forward type was used in the experiments. Its structure was defined as follows:

- number of neurons in the initial layer is equal to the number of elements in the feature vector
- number of neurons in the hidden layer is twice as large as the number of neurons in the initial layer
- each neuron in the output layer matches a different class of instruments, thus the number of neurons in the output layer is equal to the number of the classes of instruments

- neurons in the initial and the output layers have log-sigmoid transfer functions, while neurons in the hidden layer have tan-sigmoid transfer functions.

The initial stage of experiments started with the training phase of a neural network. Vectors of parameters were randomly divided into two sets: training and validation vectors. Each set contained 50% of all vectors, which meant that the representation of some instrument classes may be small whether others more numerous. The error back-propagation algorithm was used to train the neural network. The process of training was considered as finished when the value of the cumulative error of network responses for the set of testing vectors had dropped below the assumed threshold or when the cumulative error of network responses for the validation set of vectors had been rising for more than 10 consecutive cycles. The recognized class of the instrument was determined by the highest value of the output signals of neurons in the output layer. The training procedure was repeated 10 times and the best-trained network was chosen for further experiments.

In addition to the single neural network algorithm characterized above, a two-stage algorithm using a group of four neural networks was implemented (Fig. 3.38).

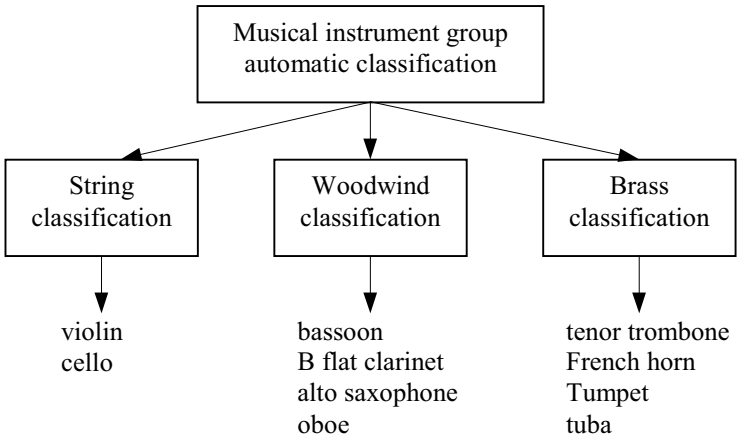


Fig. 3.38. Diagram of the two-stage musical instrument classification algorithm

The aim of the first network was to identify the group of instruments (strings, woodwinds or brass), to which the sound being analyzed belongs In the second stage, based on the response of the first network, the sound is recognized by one of the three remaining networks specialized in the classification of different groups of instruments. Each neural network used in

the experiments complies with the rules of the structure and the training process described above for a single network.

Detailed results of musical sound classification achieved by using two artificial neural network algorithms are presented in Tables 3.7-3.9.

Table 3.7. Effectiveness of the single neural network algorithm

Musical instrument	No. of samples	No. of errors	Effectiveness [%]
bassoon	179	5	97.2
B flat clarinet	195	27	86.2
oboe	173	21	87.9
tenor trombone	166	11	93.4
French horn	166	23	86.1
alto saxophone	124	6	95.2
violin	182	13	92.9
trumpet	138	5	96.4
F flat tuba	159	4	97.5
cello	214	16	92.5
Total	1696	131	92.3

Table 3.8. Effectiveness of the two-staged neural network algorithm

Musical instrument	No. of samples	No. of errors	Effectiveness [%]
bassoon	189	8	95.8
B flat clarinet	189	7	96.3
oboe	165	14	91.5
tenor trombone	166	12	92.8
French horn	163	14	91.4
alto saxophone	119	9	92.4
violin	189	19	90.0
trumpet	142	11	92.3

Table 3.8. (cont.)

F flat tuba	161	3	98.1
cello	214	10	95.3
Total	1697	107	93.7

Table 3.9. Effectiveness of the first stage of the complex neural network algorithm

Musical instrument group	No. of samples	No. of errors	Effectiveness [%]
strings	403	23	94.3
woodwinds	662	26	96.1
brass	632	19	97.0
Total	1697	68	96.0

The results of classification are better for the two-staged neural network algorithm despite the fact that an audio sample has to be recognized by two neural networks, thus their errors cumulate. Higher accuracy of classification is still possible because a 96% effectiveness of instrument group classification and a 97.6% effectiveness of instrument classification result in a total effectiveness of 93.7% for the two-staged neural networks algorithm, which is higher comparing to the single neural network algorithm.

For the single neural network algorithm, the only instruments of the accuracy of recognition lower than 90%, are clarinet and oboe. The two instruments were confused with each other. The second algorithm often incorrectly classifies sounds of violin, oboe and French horn. It is worth noticing that the results of classification of each instrument sounds are more uniform, i.e. there is a smaller difference between the best and the worst classified instrument than for the single neural network algorithm.

It can be seen that MPEG-7-based low-level audio descriptors are very suitable for the automatic musical sound classification (Kim et al 2003). Parameters derived directly from the audio spectrum (i.e. Audio Spectrum Descriptors) seem to be the most significant ones. Moreover, they are much more universal than Timbre Descriptors because they may be used to classify any type of musical sounds. Among Audio Spectrum Descriptors, the simplest parameter seems to be the most important one. That is the Audio Spectrum Envelope descriptor, which consists of coefficients describing power spectrum density in the octave bands.

Shown above, is an example of experiments that were carried out in the Multimedia Systems Department. A different configuration of descriptors contained in feature vectors along with a different configuration of neural networks were tested in many experiments (Szczuko et al 2004; Kostek et al 2004; Kostek 2004b). Typically, the smaller number of instruments to be classified, the better accuracy of classification, however in most experiments the results obtained while employing a neural network as a classifier, were clearly above 90%.

3.4 Rough Set-Based Classifier

3.4.1 Rough Sets

The first pioneering papers on rough sets, written by the originator of the idea, Professor Zdzislaw Pawlak, were published in the early eighties (Pawlak 1982). The rough set theory and its basic concepts proposed by Pawlak provide an effective tool for extracting knowledge from database (Bazan et al 1994; Bazan et al 2003; Pawlak 1996, 1998, 2003, 2004; Polkowski and Skowron 1998a; Skowron 1994a, 1994b; Slowinski et al 1996; Ziarko 1996). Since 1982, many researchers have introduced rough set theory to different scientific domains (Chmielewski and Grzymala-Busse 1994; Grzymala-Busse et al 2004; Polkowski and Skowron 1998b, 1998c; Skowron 1994b; Ziarko 1993, 1994). This theory has also been successfully utilized in the field of acoustics (Czyzewski and Kaczmarek 1993, 1994, 1995; Czyzewski et al 2004; Czyzewski and Kostek 1998; Czyzewski and Krolikowski 1998; Kostek 1996, 1997, 1998a, 1999, 2003). Rough set-based decision systems are often employed for finding hidden, implicit rules forming the basis for the experts' decisions. Such processes of extracting knowledge from data sets are known as *knowledge discovery* and *data mining*. Since the basis of rough sets is extensively covered in the literature, this would be an outline of only the general concepts.

A fundamental principle of a rough set-based learning system is the need to discover redundancies and dependencies between the given features of a problem to be classified. Several important concepts include such notions as *Upper Approximation*, *Lower Approximation* and *Boundary Region* (Fig. 3.39). The notion of approximation is a focal point in many approaches to data analysis based on rough set theory. In the majority of rough set applications the approximations are used only at some

stage of inductive learning. Most of existing solutions make use of decision rules derived from data (Pawlak 1982).

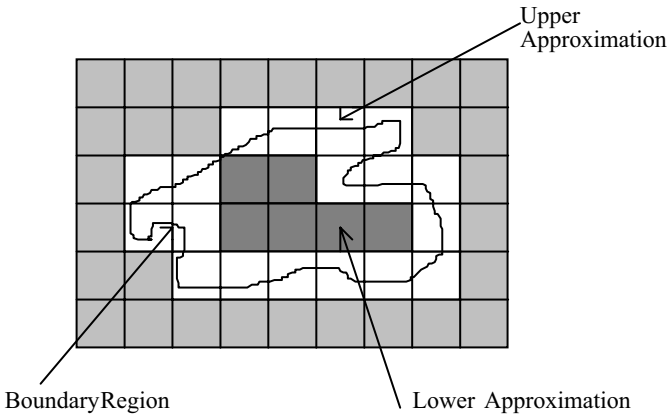


Fig. 3.39. Basic structure of rough sets

A *Universe U* is defined as a collection of objects standing at the top of the rough set hierarchy. On the other hand, a basic entity is placed at the bottom of this hierarchy. Between them, the *Approximation Space* is defined. The *Approximation Space* is partitioned by the minimum units, called equivalence classes, or also elementary sets. Lower and upper approximation definitions are based on the approximation space. Consequently, a *rough set* approximates a given concept from below and from above, using both lower and upper approximations. Three other properties of rough sets defined in terms of attribute values are shown in Fig. 3.40, namely: *dependencies*, *reduct* and *core* (Pawlak 1982).

In Fig. 3.41, the relationship between the *Universe* and the *Approximation Space* is presented. The circles represent the objects in a universe. The grid over the circles corresponds to the *Approximation Space*, which is by definition a partitioned universe.

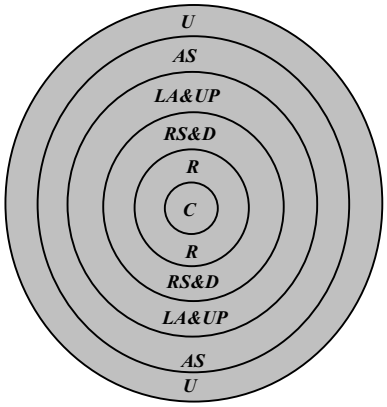


Fig. 3.40. Hierarchy of concepts in rough sets

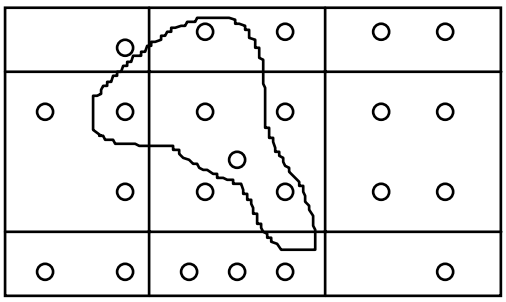


Fig. 3.41. Relationship between *Universe* and *Approximation Space*

Knowledge is represented in rough sets by a tuple $S_R = \langle U, P, D, V_P, V_D, F \rangle$. The variables are defined as follows: U is a finite collection of objects; P is a finite set of condition features or attributes; D is the decision attribute, arbitrarily chosen by an expert; V_P is the union of all condition attributes in P; V_D represents the domain of the decision attributes; and F is called a knowledge function. This means that the knowledge in rough set theory can be represented as a *Decision Table* (Pawlak 1982, 1996, 1998). A row in the *Decision Table* represents an object in the *Universe,* and each column corresponds to an attribute in P. The decision attribute is always in the very last column. Such a way of presenting knowledge is shown in Table 3.10. Information stored in a tabu-

lated form allows for distinguishing conditional attributes (premises) and decision attributes. This approach stores knowledge acquired by one or more experts in the form of IF-THEN rules (see Eq. 3.123). However, the experts' conclusions may differ, and therefore the information table becomes inconsistent resulted in possible rules. However, a major advantage of a rough-set-based decision system is the capability to process contradictory rules. On the other hand, these systems work on discrete values. If the input data are continuous, the preprocessing step must include discretization. Due to their significance, the questions associated with data discretization for the needs of rough-set-based reasoning constitute an individual field of studies, as it turns out that quantization method influences the quality of the system functioning. This issue will be presented further on.

In Fig. 3.39, the *Approximation Space* S is divided by S into three *discernibility* regions: the positive region (dark gray), the boundary region (white) and the negative region (surrounding area - gray). Assume that $R \subset U \times U$ is an equivalence relation on U which partitions U into many equivalence classes, R is called the *indiscernibility* relation. The *Lower Approximation* ($\underline{R}(S)$) of S in S is denoted as the union of the elementary sets whose members are all in S, and the *Upper Approximation* ($\overline{R}(S)$) is defined as the union of the elementary sets that have at least one member belonging to S. Resulting from these considerations, a standard set S can be approximated in space S by the pair $\underline{R}(S), \overline{R}(S)$, called the *rough set* (Pawlak 1982, 1996).

Table 3.10. Knowledge base representation in the rough set theory

object/attribute	A_1	A_2	A_3	.....	A_m	D (decision)
t_1	a_{11}	a_{12}	a_{13}	.....	a_{1m}	d_1
t_2	a_{21}	a_{22}	a_{23}	.....	a_{2m}	d_2
t_3	a_{31}	a_{32}	a_{33}	.....	a_{3m}	d_3
.....	.....	.....	.....	.....	.....	.....
t_n	a_{n1}	a_{n2}	a_{n3}	.....	a_{nm}	d_n

Rough set theory integrates a generalized approach to data, and relies on experts' knowledge about the problems to be solved. The rough set method also provides an effective tool for extracting knowledge from databases. The first step in data analysis based on the rough set theory is the creation of a knowledge-base, classifying objects and attributes within the created decision tables. Then, the knowledge discovery process is initiated in order

to remove some undesirable attributes, followed by the generalization of the concepts of desirable attributes. The final step, called reduct, is to analyze the data dependency in the reduced database and to find the minimal subset of attributes.

Decision rule is a formula of the form:

$$(a_{ij} = v_1) \wedge ... \wedge (a_{im} = v_m) \Rightarrow (decision = d) \tag{3.123}$$

where $1 \le i_1 < ... < i_m \le |P|, v_i \in V_{ai}$

Atomic formulas (a_{ij}=v_j) are called descriptors. A rule r is applicable to the object, or alternatively, the object matches rule, if its attribute values satisfy the premise of the rule. There are some numerical characteristics connected to the rule, namely matching and support.

- *length*(r) - the number of descriptors in the premise of r
- [r] – carrier of r, i.e. the set of objects satisfying the premise of r,
- *support*(r)=card([r]) - the number of objects satisfying the premise of r,
- *confidence*(r) - the measure of truth of the decision rule:

$$confidence(r) = \frac{card([r] \cap CLASS_k)}{card([r])} \tag{3.124}$$

No universal method of extracting knowledge from data exists. The existing methods exploit properties of rough sets in various ways. In rich bibliography one can find information on existing rough set-based reasoning systems, e.g. LERS, ROSETTA, RSES, PRIMEROSE, as well as descriptions of numerous algorithms for such reasoning (Bazan and Szczuka 2001; Bazan et al 2002; Chmielewski et al 1993a, 1993b; Czyzewski and Kaczmarek 1993; Grzymala-Busse and Lakshmanan 1992; Grzymala-Busse and Grzymala-Busse 1994; Lenarcik and Piasta 1994; Polkowski and Skowron 1998c; Skowron 1994a, 1994b; Tsumoto et al 1998; Ziarko 1993,1994; http://logic.mimuw.edu.pl/~rses/; http://www.rsds.wsiz.rzeszow.pl/rsds.php). The number of minimal consistent decision rules for a given decision table can be exponential with respect to the size of the decision table. Therefore some heuristics may be implemented in rough set-based systems:

- exhaustive algorithm which realizes the computation of object oriented reducts, such a method is based on Boolean reasoning approach,
- genetic algorithm along with permutation encoding and special crossover operator which allow for computing a predefined number of minimal consistent rules,

- covering algorithm which searches for minimal set of rules covering the whole set of objects,
- LEM2 algorithm which is also a covering algorithm.

In addition the support of decision rules can be increased by means of discretization, rule shortening (i.e. removal of some descriptors from a given decision rule can increase its support but at the same time it decreases its confidence), and generalization of rules by merging rules containing common descriptors.

The LERS system developed by Grzymala-Busse, uses two different approaches to rule induction, machine learning and knowledge acquisition, based on algorithms known as LEM1 and LEM2 (Learning from Examples Modules) (Chmielewski 1993a; Tsumoto et al 1998). The first algorithm is based on the global attribute covering approach, while the latter is local. LERS first checks the input data for consistency, after which lower and upper approximations are computed for every concept.

Another system based on rough set theory is the experimental KDD system designed at the University of Madrid, called RSDM, which provides a generic data mining engine (Tsumoto et al 1998). This system evolved from a previously engineered system called RDM-SQL. The system kernel includes the following modules: User Communication Module, Working Area, Dynamic Operator Loader, Mining Data Module and DW Communication Module. Another algorithm, namely TRANCE, described by its author as a Tool for Rough Data Analysis, Classification and Clustering, generates rough models of data (Tsumoto et al 1998). These models consist of a partition of the data set into a number of clusters, which are then labeled with decisions. The system uses either systematic or local search strategies. The ProbRough system is used for inducing rules from data. First, it tries to find an optimal partition of the condition attribute value space that minimizes the average misclassification cost, and then it induces the decision rules.

One of the best developed systems based on rough set theory is the ROSETTA software, which is a system for knowledge discovery and data mining (Tsumoto et al 1998). The kernel of this system was developed by the Skowron's research group at the University of Warsaw. A Norwegian group within the framework of a European project supported the GUI (Graphical User Interface) of this system. The system consists of several algorithms, the main ones of which are as follows: preprocessing of data tables with missing values, filtering of reducts and rules according to specified evaluation criteria, classification of new objects, and computing rough set approximations. The ROSETTA system provides heuristics for

search and approximations based on resampling techniques and genetic algorithms.

Rough Set Exploration (RSES) system has been created at the Warsaw University to enable multi-directional practical investigations and experimental verification of research in decision support systems and classification algorithms, in particular of those with application of rough set theory (Bazan and Szczuka 2001; http://logic.mimuw.edu.pl/~ rses/). First version of RSES and the library RSESlib was released several years ago (Bazan and Szczuka 2001; Bazan et al 2002). After modifications and improvements it was used in many applications. The RSESlib was also used in construction of the computational kernel of the ROSETTA system for data analysis (Tsumoto et al 1998).

Another system which appeared recently is ROSE (Rough Set Data Explorer), developed at the Poznań University of Technology. This system is a successor of the RoughDas and RoughClass systems which worked under the DOS operating system. ROSE is a modular program (Windows environment) which allows for performing standard and extended rough set-based analyses of data, extracting characteristic patterns from data, inducing decision rules from sets of learning examples, evaluating the discovered rules, etc. Additionally, it contains a module which offers both automatic and user-defined discretization. RSL (Rough Set Library), on the other hand, implemented at the Warsaw University of Technology, is intended as a kernel for any software implementation based on rough set theory. It offers two possible applications which may be based on an RS library, one of which is an interpreter of queries for the information system and the other of which is an expert system with a knowledge acquisition module (Tsumoto et al 1998).

An environment for the synthesis and analysis of concurrent models based on rough set theory and Petri nets, ROSEPEN, was created by a research group from Rzeszow University, PL. This system was developed using separate modules, one of which allows for handling data tables according to rough set theory. The group of Rzeszow University also created a website in which a vast number of papers related to rough sets can be found (http://www.rsds.wsiz.rzeszow.pl/rsds.php).

The RoughFuzzyLab system was engineered by a scientific group from the San Diego State University. It uses two approaches for data mining and rule extraction: one is based on rough set theory (minimum concept description), and the other uses fuzzy methodology. The PRIMEROSE (Probabilistic Rule Induction Methods based on Rough Sets) generates probabilistic rules from databases. The system is aim-oriented, specifically intended for use with medical databases. It allows not only for inducing

knowledge from data, but also provides estimation of probabilities and test statistics, cross-validation, etc. (Tsumoto et al 1998).

KDD-R (Knowledge Discovery in Data using Rough Sets) is a system developed by Ziarko (Tsumoto et al 1998). It is an extension of previously introduced systems called DataQuest and DataLogic. The basic underlying methodology behind this software-based system is rough set theory. The major components of the system consist of data preprocessing and rule search. One of the main features of this system is its ability to extract rules from data, both numerical and categorical. Also, a rough set-based rule induction algorithm was engineered at the Gdansk University of Technology (Czyzewski and Kaczmarek 1993; Czyzewski and Krolikowski 1998).

Other algorithms and systems based on rough set theory which work in different software environments and which were created at various universities for different purposes are also in existence, but they will be not cited here because they are still under development or its applications are known less widely.

Recently, the first volume of a new journal, Transactions on Rough Sets was prepared (Grzymala-Busse et al 2004). This journal, a new subline in the Springer-Verlag series Lecture Notes in Computer Science, is devoted to the entire spectrum of rough set related issues, starting from logical and mathematical foundations of rough sets, through all aspects of rough set theory and its applications, to relations between rough sets and other approaches to uncertainty, vagueness, and incompleteness, such as fuzzy sets, theory of evidence, knowledge discovery, data mining and intelligent information processing, etc. This very first volume, of which the author is co-editor, is dedicated to the mentor of rough sets, Professor Zdzislaw Pawlak, who enriched this volume with his contribution on philosophical, logical, and mathematical foundations of rough set theory. In this paper Pawlak shows the basic ideas of rough set theory as well as its relations with Bayes' theorem, conflict analysis, flow graphs, decision networks, and decision rules (Grzymala-Busse et al 2004).

3.4.2 Discretization

Feature vectors obtained as a result of the parametrization process can directly feed the inputs of classification systems, such as for example artificial neural nets, even if they consist of real values. On the other hand, rough set-based classification system requires discretized data. Some systems (e.g. RSES) incorporate discretization algorithms into the system kernel, while others need real values to be quantized. During the training phase, a number of rules are produced, on the basis of which the classifica-

tion is then performed. Since the rules produced contain parameter values, their number should thus be limited to a few values. Otherwise, the number of rules generated on the basis of continuous parameters will be very large and will contain specific values. For this reason, the discretization process is needed (Swiniarski 2001). After the discretization process is finished, parameters no longer consist of real values.

Some researchers assign discretization techniques into three different axes: global vs. local, supervised vs. unsupervised, and static vs. dynamic. The distinction between global and local methods stems from discretization when it is performed. Global discretization involves discretizing all continuous parameters prior to induction. They simultaneously convert all continuous attributes. In contrast, local methods, on the other hand, are applied during the induction process, where particular local regions may be discretized differently. They operate on a single continuous attribute at a time. Supervised methods are referred to as the ones that utilize class labels during the discretization process. Many discretization techniques require a parameter, k, indicating the maximum number of intervals to produce in discretizing a feature. Static methods perform one discretization pass of data for each feature and determine the value of k for each feature independent of the other features. Dynamic methods conduct a search through the space of possible k values for all features simultaneously, thereby capturing interdependencies in feature discretization (Dougherty et al 1995; Swiniarski 2001).

The parameter domain can be divided into subintervals and each parameter value belonging to the same subinterval will take the same value (quantization process); or parameter values can be clustered together into a few groups, forming intervals, and each group of values will be considered as one value (clusterization process).

Several discretization schemes were reviewed by Chmielewski and Grzymala-Busse (Chmielewski and Grzymala-Busse 1994), among them: Equal Interval Width Method, Equal Frequency per Interval Method, and Minimal Class Entropy Method. They also proposed a method which uses a hierarchical cluster analysis, called Cluster Analysis Method (Chmielewski and Grzymala-Busse 1994). They discussed both local and global approaches to discretization problems. This last method should be classified as global, thus producing partitions over the whole universe of attributes (Skowron and Nguyen 1995). More recently, hybrid procedures (containing both rough set and Boolean reasoning of real value attribute quantization) were proposed by Skowron and Nguyen, explaining the nature of quantization problems with respect to the computational complexity. Using this approach, further development seems promising when using

proposed methods as evaluation tools for unseen object classification (Skowron and Nguyen 1995; Nguyen 1998; Nguyen and Nguyen 1998).

Discretization methods based on fuzzy reasoning also appeared in literature (Bezdek et al 1987; Hong and Chen 1996; Kosko 1997; Kostek 1998b, 1999; Slowinski 1994). One may find methods that substitute crisp discretization subintervals with fuzzy subintervals defined by the attribute domains (Slowinski 1994). These fuzzy subintervals have overlapping boundaries which are characterized by decreasing membership functions. Following the proposal made by Slowinski et al. (1996), first some discretization methods such as minimal entropy per interval, median cluster analysis and discrimination-based method were used in the experiments. Next, for each cut point c on the attribute domain, two consecutive subintervals were defined while the heuristic approach which minimizes the information entropy measure was applied. The applied measure was used in order to check whether consecutive crisp subintervals may be substituted with adequate fuzzy subintervals (Slowinski 1994).

3.4.3 Application of Rough Sets to Musical Instrument Sound Classification

For the rough set-based classification, many experiments were carried out in the Multimedia Systems Department. The author's papers in 90s started these experiments (Kostek 1995, 1996, 1998a, 1998b, 1998c, 1999). They were carried out for the purpose of searching for feature vectors that could optimally describe a musical sound. The extracted sound attributes were based on the Fourier analysis, and resulted in a dozen of parameters (Kostek and Wieczorkowska 1996, 1997; Wieczorkowska 1999a). Also, some comparison between classifier effectiveness was performed. Especially valuable was an analysis of reducts that pointed out significant sound descriptors. Obtained decision rules also illustrate significance of particular parameters. In case of most musical instrument classes, a few attributes are sufficient to classify the investigated sound. Some of these works were done with cooperation of Wieczorkowska, who later published a series of papers that contained results obtained within her Ph.D. work, submitted at the Multimedia Systems Department (former Sound Engineering), GUT (Wieczorkowska 1999a, 1999b). She used both FFT- and wavelet-based analysis for parameter derivation. It occurred that for parameterization based on the Fourier analysis, parameters calculated for the whole sound appeared most frequently in the reducts. Both onset and quasi-steady state could be found in reducts. However, the most distinctive attributes were such as for example: a fundamental frequency, an approxi-

mate fractal dimension of a spectral graph, duration of the quasi-steady state and the ending transient, and velocity of fading of the ending transient, respectively. Especially important were parameters describing relative length of sound parts, since they enabled the recognition of pizzicato sounds. In the case of the wavelet-based parameterization attributes referring to the position of the center of the onset and the center of the steady state played a similar role to the mentioned above parameters (Wieczorkowska 1999a). Lately, some recent results were published by her and her co-workers (Wieczorkowska and Czyzewski 2003; Wieczorkowska et al, 2003).

Another set of papers which deals with rough set-based classifiers was published by Lukasik and her co-workers from the Poznan University of Technology (Lukasik 2003a, 2003b; Lukasik and Susmaga 2003; Jelonek et al 2003, 2004). However, these studies were devoted to violin timbre quality classification. They applied a mixture of unsupervised learning and statistical methods to find and illustrate the similarity and dissimilarity factors in the timbre of violin voices. They constructed the AMATI sound database that contained digitized recordings of 70 musical instruments presented at the Henryk Wieniawski 10th International Violinmakers Competition in Poznan, Poland, 2001, thus 17 000 sound files gathered in the database. Instruments whose recordings are contained in the AMATI database, are of master quality and they represent international schools of violinmaking. The database contains harmonic-based parameters for each sound. Signal waveforms, spectra and spectrograms are also available in the AMATI database. It includes open string bowed and pizzicato sounds, the entire range of notes across a chromatic scale on each string, a range of notes of diatonic scale and a fragment of J.S. Bach's work. The sound data served to extract various sets of features, including harmonic based parameters (e.g. brightness, Tristimulus, even and odd harmonics content), spectral parameters (e.g. energy, moments of various order), mel- and linear-scale cepstral coefficients, spectral envelope features (maxima and minima) and human ear auditory model features.

The collection of sounds comprises material similar to the one that the jury of musicians examined during the audition. In the analyses, Jelonek and Slowinski, other authors working with the AMATI database (Jelonek et al 2004), were interested in reconstructing the relationship between some pre-defined characteristics of the instruments and the verdict reached by the jury. On the basis of this ranking they attempted to infer a preference model that is supposed to re-construct the preference of the jury. For this purpose they used inductive supervised learning methods that include a preference-modeling tool called Dominance-based Rough Set Approach. The analysis started with constructing a rough approximation of the pref-

erence relation underlying the final ranking. This allowed inducing decision rules in terms of criteria considered by the jury but also in terms of other criteria, including various acoustic characteristics of violins. Both approaches served for discovering subsets of acoustic features that are relevant to producing rankings of violins (Jelonek et al, 2003, 2004).

Experiments at Multimedia Systems Department

Recent studies done at the Multimedia Systems Department brought some new results. Experiments were devoted to testing MPEG-7- and wavelet-based sound descriptors. For the purpose of automatic classification based on the rough set theory the Rough Set Exploration System (Bazan and Szczuka 2001) was employed. The main principles of experiments were as while using a neural network-based classifier (see Chapter 3.3). The whole set of instrument recordings was divided into learning and testing sets in 50/50 rate. The same musical instrument set was used. All MPEG-7-based attributes, exploited in previous analysis with neural networks (see Chapter 3.3), were used in the decision table (Table 3.11). In addition, in some experiments some wavelet-based attributes were added. It was found that Daubechies filters (2nd order) have the computational load considerably lower than while employing other types of filters, therefore they were used in the analysis. For the purpose of the study several parameters were calculated. They were derived by observing both energy and time relations within the wavelet subbands. Energy-related parameters are based on energy coefficients computed for the wavelet spectrum subbands normalized with regard to the overall energy of the parameterized frame corresponding to the starting transient. On the other hand, time-related wavelet parameters refer to the number of coefficients that have exceeded the given threshold. Such a threshold helps to differentiate between 'tone-like' and 'noise-like' characteristics of the wavelet spectrum (Kostek and Czyzewski 2001b). Wavelet-based parameters were as follows: cumulative energy (E_{cn}) related to nth subband, energy of nth subband (E_n), e_n - time-related parameter allowing for the characterization of the wavelet pattern, calculated for each wavelet spectrum subband, and referring to the number of coefficients that have exceeded the given threshold and f_n - variance of the first derivative of the absolute value of the wavelet coefficient sequence. By means of Fisher statistic the number of wavelet-coefficient was reduced from 50 to a few ones forming the following feature vector: $\{E_5, E_6, E_8, E_{10}/E_9, e_6, e_7, e_8, e_{10}, f_7, f_{10}, E_{c7}, E_{c8}, E_{c10}\}$.

First, real values of attributes in the learning set were discretized (Fig. 3.42a) to obtain lower number of different values. Each attribute value is

represented as a subinterval. Induced rules operate only on subintervals, giving chance to generalize the problem and to classify unknown objects. Subinterval ranges are obtained as a result of discretization algorithm. MD-heuristic is used for searches in the attribute domain cuts, which discern largest number of pairs of objects (Bazan and Szczuka 2001; Bazan et al 2002). The same cuts are then used to discretize attribute values of testing objects. Global discretization method was applied first. Next, the generation of rules was performed (Fig. 3.42b). RSES system uses genetic algorithm to induce rules (Bazan and Szczuka 2001; http://logic.mimuw.edu.pl/~ rses/).

Table 3.11. Format of the decision table

Key-Num	h_{ev}	*LAT*	*HSC*		ASE_4	ASE_5		SFM_{22}	SFM_{24}	*Decision*
57.64	0.9272	0.1072	914	...	-0.1761	-0.1916	...	0.0971	0.0707	cello
57.68	0.9178	0.1140	818		-0.1634	-0.1727		0.0927	0.0739	cello
...	...	...	...	...	...	...	...	...	...	...
53.03	0.7409	-0.7958	875		-0.2115	-0.2155		0.1108	0.0775	t. trombone

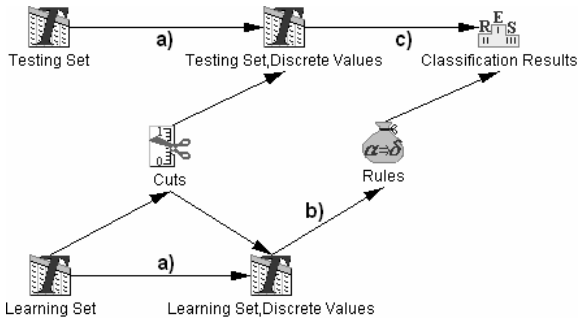


Fig. 3.42. Decision system description, a) discretization, b) generation of rules, c) classification (Bazan and Szczuka 2001)

Each rule has an implication form, conditioning a decision on attribute values:

$[ASE_9 \in (-\infty,-0.15525) \wedge ASE_{10} \in (-0.16285,+\infty) \wedge ASE_{11} \in (-0.16075,+\infty) \wedge ASE_{13} \in (-0.18905, +\infty) \wedge ASE_{26} \in (-\infty,-0.18935)] \Rightarrow$ [decision = violin]

or:

IF [(a value of ASE_9 belongs to the interval (-∝,-0.15525)) AND (a value of ASE_{10} belongs to the interval (-0.16285,+ ∝)) AND ... AND (a value of ASE_{26} belongs to the interval (-0.18935,+ ∝))] THEN [decision IS violin]

Rules are used to classify unknown objects of the testing set (Fig. 3.42c). First, attributes of a new object are discretized with required cuts, and then, all rules are applied. If more that one rule is matching the object, then final decision is based on a voting method.

Over 18000 rules were induced in the tests. Their principle was to classify a set of ten musical instruments. Most of them covered only few cases (Fig. 3.43). Maximum length of an induced rule (i.e. a number of attributes in an implication) is 11, minimum is 3, and the average is 6 (Fig. 3.44).

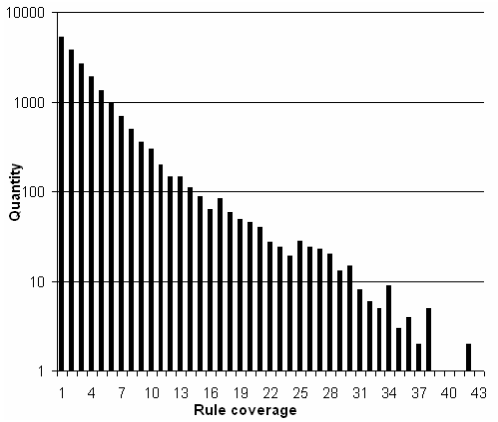


Fig. 3.43. Quantity of rules covering the given number of cases

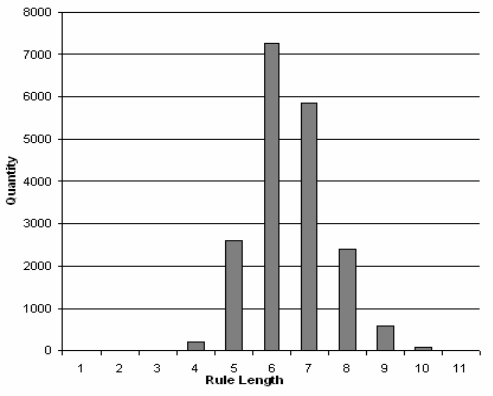


Fig. 3.44. Histogram of the rule length

The results of classification employing a rough set-based system are presented in Table 3.12. The RSES system tested FVs that contained MPEG-7 based descriptors.

Table 3.12. Classification accuracy while applying a global discretization method

Musical instrument	No. of samples	No. of errors	Effectiveness [%]
bassoon	201	60	70.1
B flat clarinet	184	32	82.6
oboe	165	12	92.7
tenor trombone	171	25	85.4
French horn	166	28	83.1
alto saxophone	119	19	84.0
violin	208	45	78.4
trumpet	137	8	94.2
F flat tuba	137	13	90.5
cello	205	37	82.0
Total	1693	279	83.5

The average classification accuracy was 84%. The worst results were obtained for a bassoon and a violin. The instruments belonging to strings were often confused with each other.

Another experiment consisted in discretization that employed a local method. Change of the experiment principles regarded FVs content. FVs contained a joint representation of parameters, namely MPEG-7 and wavelet-based descriptors. To this end data were first divided into 50%/50% training and testing sets (1695 samples in each set). The length of a rule was limited to 10. In total, 27321 rules were derived. The average accuracy was 89% (see Table 3.13). The denotations in Table 3.13 are as follows: a bassoon (BAS), a B flat clarinet (CL), an oboe (OB), a tenor trombone (TT), a French horn (FH), an alto saxophone (AS), a violin (VI), a trumpet (TR), a F flat tuba (TU), a cello (CE), a coverage (Cov). Results for the combined representation are given by the diagonal of Tab. 3.13 (number of sounds correctly recognized). Other values denote the system erroneous answers made in the recognition process. The overall accuracy is also visible in this table.

It is also interesting to apply a rough set-based system in the context of a feature vector redundancy search. As expected, reducts obtained in the analysis consist of a fraction of the whole feature vector. For the MPEG-7- and the wavelet-based feature vectors, the following parameters were found significant: *HSD*, *TC*, E_8, *KeyNumber*, *SC*, E_5, ASE_{14} and E_{c10} in the sense that they were the most frequent parameters used in rules.

Table 3.13. Classification accuracy while applying a local discretization method (1695 samples in training and testing sets)

	BAS	CL	OB	TT	FH	SA	VI	TR	TU	CE	No. of obj.	Accuracy [%]	Cov.
BAS	162	0	2	1	4	3	2	0	10	2	186	0.871	1
CL	0	168	12	0	0	1	1	3	2	0	187	0.898	1
OB	0	2	147	0	0	2	10	3	0	0	164	0.896	1
TT	4	0	1	146	12	3	0	2	8	1	177	0.825	1
FH	11	3	0	19	120	0	0	2	8	0	163	0.736	1
SA	0	0	1	0	0	118	1	4	0	1	125	0.944	1
VI	2	4	6	0	0	1	172	2	0	6	193	0.891	1
TR	0	0	0	1	0	6	0	138	0	0	145	0.952	1
TU	0	0	0	3	0	0	0	0	145	0	148	0.98	1
CE	0	3	1	0	0	4	11	0	0	191	210	0.91	1

Descriptors that were used in rule derivation are gathered in Table 3.14, other descriptors were disregarded while creating a set of rules of this classification task.

Table 3.14. Descriptors used in rule derivation

1	2	3	4	5	6	7	8	9	10
KeyNum	*Br*	h_{ev}	*LAT*	*TC*	*SC*	*HSC*	*HSD*	*HSDv*	*HSS*
11	12	13	14	15	16	17	18	19	20
HSV	ASE_1	ASE_3	ASE_4	ASE_5	ASE_7	ASE_8	ASE_9	ASE_{10}	ASE_{12}
21	22	23	24	25	26	27	28	29	30
ASE_{14}	ASE_{15}	ASE_{16}	ASE_{19}	ASE_{24}	ASE_{26}	ASE_{28}	ASE_{31}	ASE_{32}	ASE_{33}
31	32	33	34	35	36	37	38	39	40
ASE_{34}	*ASEm*	$ASEv_2$	$ASEv_{12}$	$ASEv_{13}$	$ASEv_{16}$	$ASEv_{21}$	$ASEv_{23}$	$ASEv_{27}$	$ASEv_{29}$
41	42	43	44	45	46	47	48	49	50
$ASEv_{33}$	$ASEv_{34}$	*ASCv*	*ASSv*	SFM_5	SFM_8	SFM_{11}	SFM_{14}	SFM_{15}	SFM_{16}
51	52	53	54	55	56	57	58	59	60
SFM_{18}	SFM_{21}	SFM_{23}	$SFMv_5$	$SFMv_8$	$SFMv_{19}$	$SFMv_{24}$	E_8	E_{10}/E_9	E_5
61	62	63	64	65					
E_6	e_8	E_{c10}	E_{c8}	E_{c7}					

In another experiment the division ratio of training and testing samples was 2/3 (2370 training samples and 1020 testing samples). The analysis resulted in 34508 rules, and the classification accuracy reached 91.93% (see Table 3.15). FVs contained a joint representation of sound samples, namely MPEG-7- and wavelet-based descriptors.

Table 3.15. Classification accuracy while applying a local discretization method (2370 training samples and 1020 testing samples)

Musical instrument	No. of samples	Accuracy
bassoon	112	0.964
B flat clarinet	112	0.938
oboe	99	0.889
tenor trombone	106	0.858
French horn	98	0.878
alto saxophone	75	0.907
violin	116	0.905
trumpet	87	0.92
F flat tuba	89	0.989
cello	126	0.944

It was also decided that the experiment would be extended to 24 musical instrument classes. They were: alto trombone (1), alto flute (2), Bach trumpet (3), bass clarinet (4), bass trombone (5), bass flute (6), bassoon (7), Bb clarinet (8), C trumpet, (9), CB (10), cello (11), contrabass clarinet (12), contrabassoon (13), Eb clarinet (14), English horn (15), flute (16), French horn (17), oboe (18), piccolo (19), trombone (20), tuba (21), viola (22), violin (23), and violin ensemble (24) classes. Most errors resulted from similarities in the timbre of instruments, for example: such pairs of instruments as: a clarinet and a bass clarinet, a trombone and a bass trombone, and also a contrabass (CB) and a cello were often misclassified due to their timbre similarity. In overall, in the case of 24 instruments the system accuracy was equal to 0.78.

Results of musical instrument classification based on rough sets are very satisfying. The average classification accuracy is higher than 90% for a dozen of instrument classes. It must be also stressed that the algorithm operated under very demanding conditions: audio samples originated from two different sources and in most experiments only 50% of the samples

was included in the training/pattern sets. It should be also remembered that these classes contained sound samples of a differentiated articulation. Classification results are instrument-dependent. Instruments having very similar sound (e.g. tuba – trombone) or the same scale range (e.g. trombone – bassoon) were most often confused with each other.

The obtained results are close to those achieved by neural networks. However, a very essential feature of a rough set-based decision system is that the system supplies the researcher with a set of transparent rules.

3.5 Minimum Distance Method-Based Classifier

3.5.1 Nearest-Neighbor Method and *k*-Nearest-Neighbor Method

Minimum-distance methods, to which the nearest-neighbor (NNr) method and the *k*-Nearest-Neighbor (*k*-NN) method belong, are used very often due to their simple rules of operation and easy algorithm implementation.

In the NNr method first all vectors of a training set are memorized and then for the need of a new object classification, its distance from the elements of the training set is computed. Recognition produces the class to which an object which is closest to the one being analyzed belongs (see Fig. 3.45).

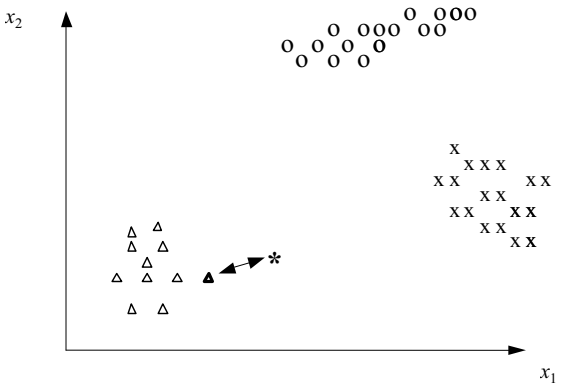


Fig. 3.45. In the case of the NN algorithm the decision-making rule assumes that the unknown object (marked with an asterisk) will be classified to the cluster of objects closest in feature space

In the *k*-NNr method *k* nearest neighbors are considered, where *k* is usually a small integer. The object being recognized is then assigned to a class, to which most of the *k* nearest neighbors belong. Such approach precludes errors resulting from mistakes in the training sequence. The idea of distance is associated with metrics defined in the property space. The appropriate metrics can be chosen empirically, in principle. This is a crucial problem of a fundamental impact on the obtained effects.

Among the most often used metrics are the following:

- Euclidean metric:

$$d(x,y)=\sqrt{(x-y)(x-y)^T} \tag{3.125}$$

where $x = [x_1, ..., x_N]$, $y = [y_1, ..., y_N]$

- generalized Euclidean metric:

$$d(x,y)=\sqrt{(x-y)W(x-y)^T} \tag{3.126}$$

$$W=\begin{bmatrix} w_1 & 0 & ... & 0 \\ 0 & w_2 & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & w_n \end{bmatrix} \tag{3.127}$$

where **W** is a diagonal weights matrix:

Normalizing factors w can be associated with the vector constituent values in various fashions, e.g. they can depend on the variability range of their values. This leads to independence of conceivable dimensional differences of individual vector constituents.

- street metrics:

$$d(x,y)=\sum_{i=1}^{n}|x_i-y_i| \tag{3.128}$$

- Hamming metrics

$$d(x,y)=\frac{1}{n}\sum_{i=1}^{n}|x_i-y_i| \tag{3.129}$$

- Canberra metrics

$$d(x,y)=\sum_{i=1}^{n}\frac{|x_i-y_i|}{x_i+y_i} \tag{3.130}$$

- Mahalanobis metrics

$$d(x,y)=\sqrt{(x-y)C^{-1}(x-y)^T} \tag{3.131}$$

where **C** is a covariance matrix of the discussed element set. Using this measure of distance leads to rectifying a non-orthogonal set of coordinates.

A disadvantage of the NNr and *k*-NN methods is the need to store all the training set elements as well as their large computational complexity. These algorithms do not possess any generalization mechanisms either. Another approach to the classification problem is the creation of pattern characteristics for each class of objects. In such a case a smaller number of objects is stored, what diminishes computational load.

These algorithms were used by the author and her team for musical object classification. Examples of such processing will be shown further on.

3.5.2 K-Means Cluster Analysis

Cluster analysis is a multivariate procedure for identifying groupings within data. The objects in these groups may be cases or variables. A cluster analysis of cases is like a discriminant analysis because it attempts to classify a set of data into groups. However, unlike in a discriminant analysis, neither the number nor the members of the groups are known. Cluster analysis is also called segmentation analysis or taxonomy analysis, because it searches to identify homogenous subgroups of cases in a population. This means identification of a set of groups, which both minimize a within-group variation and maximize a between-group variation. The first step in cluster analysis is the establishment of the similarity or the distance matrix. This matrix is a table in which both the rows and columns are the units of the analysis and the cell entries are measures of similarity or distance for any pair of cases.

K-means cluster analysis uses Euclidean distance. Initial cluster centers are chosen in a first pass of the data, then each additional iteration groups observations based on the nearest Euclidean distance to the mean of the cluster. Thus cluster centers change at each pass. The process continues until cluster means do not shift more than a given cut-off value or until the iteration limit is reached.

There are some similarity measures used in the identification of a data set. One of them is correlation. Data table in which columns are variables and rows are cases constitute the cells of the similarity matrix. Binary matching is another similarity measure, where 1 indicates a match and 0 indicates no match between any pair of cases. There are multiple matched attributes and the similarity score is the number of matches divided by the number of attributes being matched. It is usual in binary matching to have several attributes because there is a risk that when the number of attributes is small, they may be orthogonal (uncorrelated), and clustering will be indeterminate.

Another technique, called Hierarchical Cluster Analysis, attempts to identify relatively homogeneous groups of cases or variables based on selected characteristics. It uses an algorithm that starts with each case or variable in a separate cluster and combines clusters until only one is left. Hierarchical Cluster Analysis is considered as an exploratory method. Results should be confirmed with additional analysis and, in some cases, additional research. Such techniques are often applied to MIR domain.

3.5.3 Application of *k*-Nearest-Neighbor Method to Musical Instrument Sound Classification

A variety of experiments were performed based on different configurations of feature vectors and *k*-NN method used as a classifier. Also, both hierarchic and direct forms of classification were evaluated. Among others, are studies by Kaminskyj and Materka (1995), and later by Martin and Kim (1998), Fujinaga (1998), Martin (1999), Fujinaga and MacMillan (2000), Kaminskyj (2000; 2002), Eronen (2001), Agostini et al. (2001), Wieczorkowska et al. (2003) can be cited. Typically, the *k*-NN classifier outperformed other recognition systems tested on the same set of sound samples, however most authors pointed out that such a system does not provide a generalization mechanism. A comparison of results obtained by different authors is difficult to present, because apart from different parameters and a different number of musical instrument classes, and in addition, a different origin of sound samples, also a cross validation procedure was performed based on different conditions. In some studies a split of training and testing data was 70%/30%, while in others it was 50%/50%, also other validation techniques were used. Eronen in his study created a very comprehensive summary of results obtained by other authors; before proceeded his own experiments (Eronen and Klapuri 2000; Eronen 2001). In recently published studies, MPEG-7-based descriptors are often employed in experiments with *k*-NN classifiers, and in addition, for the pur-

pose of testing the generalization ability of classifiers, sound samples originating from different sources are used.

Experiments at GUT

In addition to the decision systems used in experiments carried out at the Multimedia Systems Department, described before in this Chapter, the nearest neighbor algorithm was implemented for the comparison purposes. The algorithm finds a vector from a pattern set that is the most similar (has the smallest distance) to the vector being recognized. A Hamming distance measure is used and only one nearest neighbor is taken into account. It is assumed that the sound sample being recognized belongs to the same class of instruments as the nearest neighbor. The pattern set and the validation set contain 50% of all vectors. Vectors for the pattern set are chosen randomly. The process is repeated 10 times and the pattern sets of vectors providing the best effectiveness of the classification is retained for further experiments.

Detailed results of the musical instrument sound classification with the nearest neighbor algorithm are shown in Table 3.16.

Table 3.16. Effectiveness of the nearest neighbor algorithm

Instrument	No. of samples	No. of errors	Effectiveness [%]
bassoon	173	16	90.8
B flat clarinet	185	10	94.6
oboe	155	8	94.8
tenor trombone	164	10	93.9
French horn	179	17	90.5
alto saxophone	129	2	98.5
violin	204	12	94.1
trumpet	137	0	100
F flat tuba	158	0	100
cello	212	13	93.9
Total	1696	88	94.8

All sample sounds of trumpets and F flat tubas were recognized correctly. The worst classification effectiveness was obtained for bassoon sounds, which were confused with tenor trombones and F flat tubas, and

for French horns, which were confused with tenor trombone sounds. There is a noticeable difference between the results of the best and the worst classified instrument, reaching almost 10%.

Algorithm Comparison

The best total of the effectiveness of 94.8% has been obtained by the nearest neighbor algorithm (denotation in Fig. 3.46 is given as *k*-NN), however both neural network algorithms single and two-staged (denotations in Fig. 3.46 are: singleNN and t-sNN) are only 1-2% worse. The effectiveness of rough set-based (RS) algorithms (both using global and local discretization methods) was slightly behind the other ones, which is illustrated in Fig. 3.46.

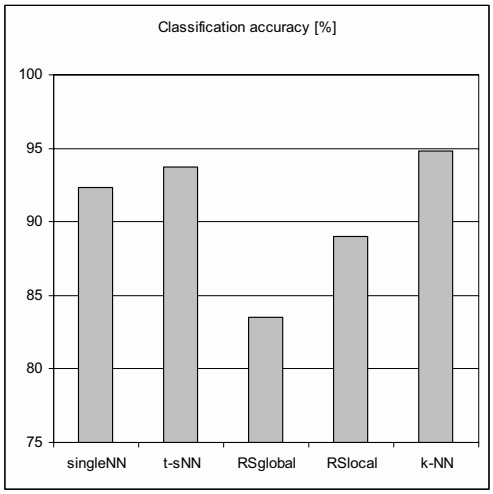


Fig. 3.46. Effectiveness of musical instrument classification

It was also checked whether, despite the algorithm used, the results of classification depend on the source of the sound samples: Catalogue or MUMS (Table 3.17).

It is clearly seen that some algorithms better identify sound samples from the Catalogue, others from MUMS. It means that the origin of the sound sample influences the result of classification, however no satisfying explanation can be given to this fact.

Table 3.17. Source of sound samples and effectiveness of classification

Algorithm	Origin of samples – classification [%]	
	Catalogue	MUMS
single neural network	92.0	93.1
two-staged neural network	94.6	90.4
rough sets	85.4	83.2
rough sets	89	87.4
nearest neighbor	94.1	97.4

Additional experiments were carried out with the number of recognized classes increased to 16. The additional instruments were: viola, bass trombone, English horn, baritone saxophone, soprano saxophone and B flat tuba. The feature vector used in experiments was the same as for 10 instruments. Summary results of classification are presented in Table 3.18.

Table 3.18. Results of 16 musical instruments classification

Algorithm	No. of samples	No. of errors	Effectiveness
single neural network	2517	340	86.5%
two-staged neural network	2517	272	89.2%
nearest neighbor	2517	238	90.5%

Extending the number of recognized instruments resulted in lowering the total effectiveness by approx. 6% for each algorithm. Still, the effectiveness of the nearest neighbor algorithm was slightly better than this of the two-staged neural network.

The additional instruments used in the experiments are very closely related to the previous ones. Practically, all errors generated by the decision systems are the results of confusing pairs of very similar instruments (e.g. viola – violin, bass trombone – tenor trombone). It is assumed that the effectiveness of classification may be improved provided that the feature vector is updated accordingly to the sound characteristics of all 16 instruments.

3.6 Genetic Algorithm-Based Classifier

3.6.1 Evolutionary Computing

As reviewed by Spears (http://www.cs.uwyo.edu/~wspears/overview/), the origins of evolutionary algorithms can be traced to at least the 1950's and 1960s (Box 1957; Fraser 1962). Over the next decades, three methodologies have emerged: evolutionary programming (Fogel et al 1966), evolution strategies (Rechenberg, 1973), genetic algorithms (Davis 1991; Holland 1975), and their applications (Banzhaf et al 1998; Goldberg 1989; Horner and Goldberg 1991; De Jong 1992, De Jong and Spears 1991; Fogel 1992; Fraser and Burnell, 1970; Koza 1991, 1992; Michalewicz 1992; Spears and De Jong 1991, Papadopoulos and Wiggins 1998; Srinivas and Patnaik 1994). The principal constituents of the evolutionary computation (EC) are: genetic algorithms (GA), evolution strategies (ES), evolutionary programming (EP), genetic programming (GP), and classifier systems (CS). Algorithms used within Evolutionary Computation are based on the principles of natural evolution used to solve a wide range of problems which may not be solvable by standard techniques. Central to such systems is the idea of a population of genotypes that are elements of a high dimensional search space. For example, in simple genetic algorithms (Goldberg 1989), genotypes are binary strings of some fixed length (n) that code for points in an n-dimensional Boolean search space.

Evolutionary computation uses the computational models of evolutionary processes as key elements in the design and implementation of computer-based problem solving systems. There are a variety of evolutionary computational models that have been proposed and studied which can be referred to as evolutionary algorithms. They share a common conceptual base of simulating the evolution of individual structures via processes of selection and reproduction. These processes depend on the perceived performance (fitness) of the individual structures as defined by an environment. More precisely, evolutionary algorithms maintain a population of structures that evolve according to rules of selection and other operators, such as recombination and mutation. Each individual in the population receives a measure of its fitness in the environment. Selection focuses attention on high fitness individuals, thus exploiting the information available on fitness. Recombination and mutation perturb those individuals, providing general heuristics for exploration. Although simplistic from a biologist's viewpoint, these algorithms are sufficiently complex to provide robust and powerful adaptive search mechanisms.

An evolutionary algorithm typically initializes its population randomly, although domain specific knowledge can also be used to bias the search. Evaluation measures the fitness of each individual according to its value in an environment. Evaluation may be as simple as computing a fitness function or as complex as running an elaborate simulation. Selection is often performed in two steps, parent selection and survival. Parent selection decides who becomes a parent and how many children the parents have. Children are created via recombination, which exchanges information between parents, and via mutation, which further perturbs the children. The children are then evaluated. Finally, the survival step decides who survives in the population.

Although similar at the highest level, each of these varieties implements an evolutionary algorithm in a different manner. The differences touch upon almost all aspects of evolutionary algorithms, including the choice of representation for individual structures, types of selection mechanisms, forms of genetic operators, and measures of performance. These approaches in turn have inspired the development of additional evolutionary algorithms such as 'classifier systems'. The interested reader is encouraged to study a very reach literature for more details (http://www.cs.uwyo.edu/~wspears/overview/).

3.6.2 Evolutionary Programming

Evolutionary programming (EP), developed by Fogel et al. (1966) traditionally has used representations that are tailored to the problem domain (http://www.cs.uwyo.edu/~wspears/overview/). For example, in real-valued optimization problems, the individuals within the population are real-valued vectors. Similarly, ordered lists are used for traveling salesman problems, and graphs for applications with finite state machines. EP is often used as an optimizer, although it arose from the desire to generate machine intelligence.

After initialization, all N individuals are selected to be parents, and then mutated producing N children. These children are evaluated, and N survivors are chosen from $2N$ individuals, using a probabilistic function based on fitness. In other words, individuals with greater fitness have a higher chance to survive. The form of mutation is based on the representation used, and is often adaptive. For example, when using a real-valued vector, each variable within an individual may have an adaptive mutation rate that is normally distributed with a zero expectation. Recombination is not generally performed since the forms of used mutation are quite flexible and can produce perturbations similar to recombination, if desired. As dis-

cussed in the later section, one of the interesting and open issues is the extent to which an EA is affected by the choice of the operators used to produce variability and novelty in evolving populations.

Evolution Strategies

Evolution strategies (ESs) were independently developed by Rechenberg (1973), with selection, mutation, and a population of size one. Schwefel (1977) introduced recombination and populations with more than one individual, and provided a nice comparison of ESs with more traditional optimization techniques. Due to initial interest in hydrodynamic optimization problems, evolution strategies typically use real-valued vector representations (http://www.cs.uwyo.edu/~wspears/overview/).

After initialization and evaluation, individuals are selected uniformly and randomly to be parents. In the standard recombinative ES, pairs of parents produce children via recombination, which are further perturbed via mutation. The number of children created is greater than N. The survival is deterministic and is implemented in one of two ways. The first one allows N best children to survive and to replace their parents. The second one allows N best children and their parents to survive. Like in EP, considerable effort has focused on adapting mutation as the algorithm runs by allowing each variable within an individual to have an adaptive mutation rate that is normally distributed with a zero expectation. Unlike in EP, however, recombination does play an important role in evolution strategies, especially in adapting mutation.

3.6.3 Genetic Algorithms

Genetic algorithms (GAs), developed by Holland (1975), combine selection, crossover, and mutation operators with the goal of finding the best solution to a given problem. A genetic algorithm is a stochastic search technique based on the principles of biological evolution, natural selection, and genetic recombination, simulating survival of the fittest in a population of potential solutions or individuals. Typically, a domain independent representation, namely, bit-strings are used. These strings are referred to as chromosomes. Sites on the chromosome corresponding to specific characteristics of the encoded system are called genes, and may assume a set of possible values, thus corresponding to alleles. Many recent applications of GAs focused on other representations, such as graphs (neural networks), ordered lists, real-valued vectors, and others. GAs are often used as optimizers, although some researchers emphasize its general adaptive capabili-

ties (De Jong 1992). Genetic algorithms are typically implemented as a computer simulation in which a population of abstract representations (chromosomes) of candidate solutions (individuals) to an optimization problem evolves toward better solutions.

The basic genetic algorithm operation (see Fig. 3.47) proceeds in the following phases (http://www.cs.uwyo.edu/~wspears/overview/):

- initiation – selecting initial chromosome population,
- fitness evaluation of chromosomes in a population,
- testing for the stop condition,
- chromosome selection,
- using genetic operators,
- creating a new population,
- output of the most fitted chromosome.

Each phase characteristics are shown below in this section.

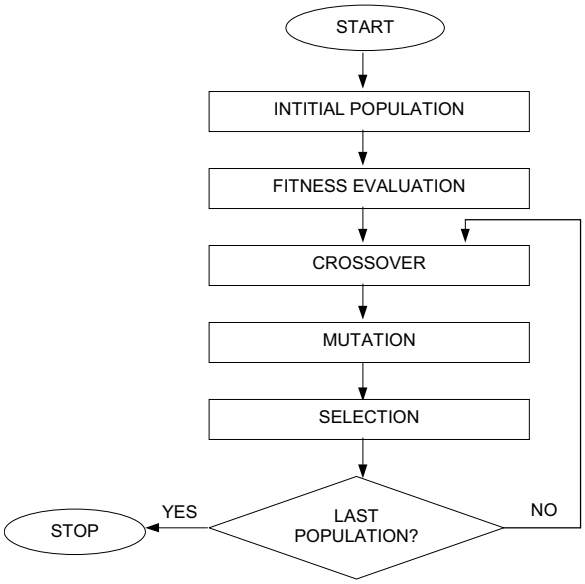


Fig. 3.47. Genetic algorithm operation flowchart

Initiation - Selecting Initial Chromosome Population

Selection of the initial chromosome population can be carried out in many ways. In a great number of cases it will be carried out randomly which is supposed to assure the uniform distribution of the initial population and –

in consequence – makes it easier to find a global optimum and prevents coming to a local extremum of the fitness function.

After the initialization parents are selected according to a probabilistic function based on a fitness criterion. In other words, the individuals with higher fitness are more likely to be selected as parents. N children are created via recombination from N parents. N children are mutated and survive, replacing N parents in the population.

Fitness Evaluation of Chromosomes in a Population

The evaluation step consists in calculating the fitness function value for each chromosome from this population. The higher the fitness value the better adapted a given chromosome. The solution can be exact (extremum of the fitness function) or approximate. The approximate solution may come from the fact that the chain quality can depend on the parameter, at which the quality evaluation is done, e.g. from the initial state in the case of control. To free from the influence of a parameter, quality calculations for many parameter values must be carried out and obtained results must be averaged.

Testing for the Stop Condition

Determining the stop condition depends on a concrete use of the genetic algorithm. In optimization methods, if the maximum (or minimum) value of the fitness function is known, stopping the algorithm can happen after the required optimal value is achieved, or with a specified accuracy. Therefore, the optimization process will be performed until some convergence criteria (the maximum fitness in the population ceases to increase) are satisfied. Stopping the algorithm can also take place, if its continued working does not improve the best value obtained so far. The algorithm can be also stopped after a certain operation time has expired or after a certain number of iterations. If the stop condition is fulfilled, then the pass to the last step happens. The output is the best fitted chromosome. Otherwise, the next step is selection.

Chromosome Selection

The chromosome selection is based on the values calculated by the fitness function. The selected chromosomes take part in producing descendants to the next generation. The selection happens according to the natural selection principle – the best chance for being selected have the chromosomes that show the highest value of the fitness function. In order to prevent the

solution set from converging too early to a sub-optimal or local solution, the selection is biased towards elements of the initial generation, which have better fitness, though usually not to such a degree that poorer elements have no chance to participate.

There are many selection methods, whereby the roulette wheel method is the most popular one. In the roulette wheel method, also called the fitness proportionate selection, the fitness level is used to associate each individual chromosome with a probability of selection. While the candidate solutions with higher fitness are not likely to be eliminated, there is still a chance that some weaker solutions may also survive the selection process. This could be an advantage, even though a solution may be weak, it may include some components, which may prove to be useful in the process of recombination. Another method, often used in GAs is the tournament selection method. It makes n tournaments in order to choose n individuals. The individual with the highest fitness in the group of k elements is selected, the others are removed. The most widely spread tournament selection is at $k=2$.

The roulette wheel method consists in assigning to each chromosome a sector of the roulette wheel, which is proportional to the fitness function level for a given chromosome. Therefore, the higher the fitness function value the bigger the sector of the roulette. The whole roulette wheel corresponds to the sum of the fitness function values of all chromosomes of a given population. Each chromosome, that is denoted by ch_i, for $i = 1, 2, ..., N$, where N is the population dimension, corresponds with the wheel sector $v(ch_i)$ that forms the fragment of the whole wheel, expressed in percentage according to the formula (Michalewicz 1992):

$$v(ch_i) = p_s(ch_i) \cdot 100\% \tag{3.132}$$

where $p_s(ch_i)$ is the selection probability of the chromosome ch_i.

The chromosome selection can be understood as the roulette wheel rotation with the consequence that the chromosome belonging to the drawn sector of the roulette wheel will be chosen. The bigger the heel sector (or higher the level of the fitness function), the higher the probability that a given chromosome will be chosen. As the result of the selection process, the parental population (a so-called parental pool) will be created, with the same dimension as the current population of N individuals.

Using Genetic Operators

Using genetic operators leads to creating a new population of descendants obtained from a parental pool chosen by a selection method. In the classical genetic algorithm, apart from selection, two basic operators are used:

- crossover (recombination),
- mutation.

Crossover takes a portion of each parent and combines the two portions to create the offspring. A number of recombination operators are widely used. The most popular are one-point (or single-point), multi-point (or *n*-point) and uniform recombinations. One-point recombination inserts a cut-point within two parents. Then the information from the segments before the cut-point is swapped between the two parents. Multi-point recombination is a generalization of this idea introducing a higher number of cut-points. Information is then swapped between pairs of cut-points. Uniform crossover does not use cut-points, but simply a global parameter to indicate the likelihood of each variable to be exchanged between two parents.

The first stage of the crossover consists in selecting a pair of chromosomes from a parental population. This is a temporary population consisting of chromosomes chosen using the selection method and assigned for further processing by means of genetic operators to create a new a population of descendants. The chromosomes from the parental population will be joined in pairs randomly, according to probability p_c (it is assumed in general that $0.6<p_c<1$). For each pair of parents – selected this way – a gene position in a chromosome will be drawn. That gene position determines a so-called crossover point. The selection of the crossover point l_k will resolve into drawing a number from the range $[1, L\text{-}1]$. As a result of the crossover of a pair of parental chromosomes, the following pair of descendants will be obtained:

- descendant whose chromosome consists of genes derived from the first parent at positions $1, ..., l_k$, and derived from the second parent at positions $l_k+1, ..., L$,
- descendant whose chromosome consists of genes derived from the first parent at positions $l_k+1, ..., L$, and derived from the second parent at positions $1, ..., l_k$.

GAs typically use mutation as a simple background operator, to ensure that a particular bit value is not lost. The mutation operator is of secondary importance in relation to the crossover operator. Mutation is needed to guard against premature convergence, and to guarantee that any location in the search space may be reached. According to the probability of mutation

p_m (in general $0<p_m<0.1$) reverses the gene value in a chromosome to an opposite one (from 0 to 1 or vice versa). Carrying out mutation according to probability p_m consists in, e.g., drawing a number from the range [0, 1] for each gene and choosing such genes for mutation for which the drawn number is lower than or equals p_m .

Creating a New Population

The chromosomes obtained as a result of a genetic operator enter into the composition of a new population. This population is called a current population for a given population of a genetic algorithm. The fitness function value for each of the chromosomes of that population will be calculated in every subsequent iteration. Thereafter, the stop condition of the algorithm will be tested and either a result in the form of a chromosome of the highest value of the fitness function will be outputted, or the pass to the selection will take place. In the classical genetic algorithm, the whole foregoing chromosome population will be replaced by a new population, as numerous as the old one.

Output of the Most Fitted Chromosome

If the stop condition of an algorithm is fulfilled, the result of the algorithm operation should be the output, in other words the solution to the problem should be produced. The best solution is a chromosome with the highest value of the fitness function.

Problems in Designing Genetic Algorithms

Central to every evolutionary algorithm is the concept of fitness (i.e., evaluation). The selection might only be based on the relative ordering of fitness. This form of selection is often referred to as ranking selection, since only the rank of individuals is of importance. All individuals are selected to be parents. Each parent is mutated once, producing N children. A probabilistic ranking mechanism chooses the best N individuals for survival, from the union of the parents and children. Again, this is a selection mechanism based on a rank. Although the GA community advocated ranking for some situations, but they also believe that fitness functions should be searched differently. Fitness proportional selection is a probabilistic selection mechanism of the traditional GA. Parent selection is performed based on how fit an individual is with respect to the population average. For example, an individual with the fitness twice the population average will tend to have twice as many children as average individuals. Survival,

though, is not based on fitness, since the parents are automatically replaced by the children.

Despite some work on adapting representation, mutation, and recombination within evolutionary algorithms, very little has been accomplished with respect to the adaptation of population sizes and selection mechanisms. The biggest problem that arises in a design phase of a genetic algorithm is finding a proper coding. Such algorithms do not show convergence with suboptimal solutions for every code. Good coding should fulfill at least two such conditions. Namely, it should be immune from the crossover operation (i.e. after the replacement of genetic material the new sequences should be available for decoding). It is a common case that efforts will be undertaken to match the crossover operation with the coding used. Secondly, small, coherent fragments of code sequences should reflect some specific features of the solutions. Only then the descendants inherit the properties of their ancestors. The next problem is selecting a suitable target function. This function is not always given explicitly in the problem, and even if it is, its modification – at maintaining the maximum – can prove to be advantageous. The key importance for the convergence of subsequent iterations is also the correct selection of relevant coefficients (crossover and mutation probability, the population size, etc.).

Genetic algorithms developed at the Multimedia Systems Department (GUT) were used in many applications, some of them will be shown in the following Section, some others later on in subsequent Chapters.

3.6.4 Application of Genetic Algorithms to Musical Instrument Sound Classification

The motivation for the presented experiments has arisen from the study of Lim and Tan (Lim and Tan 1999). The authors used genetic annealing algorithm (GAA) for optimizing sound parameters of the double frequency modulation (DFM) synthesis technique. Synthesized sound samples obtained by using DFM and GAA were very similar to sounds generated by musical instruments. Thus a question may be posed whether it is possible to use such techniques in the process of automatic identification of musical instrument sounds. The process is shown in Fig. 3.48, and the program flow chart is presented in Fig. 3.49.

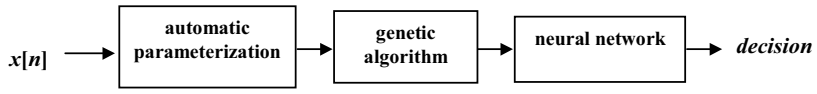


Fig. 3.48. Block diagram of the algorithm of the automatic parameter extraction system

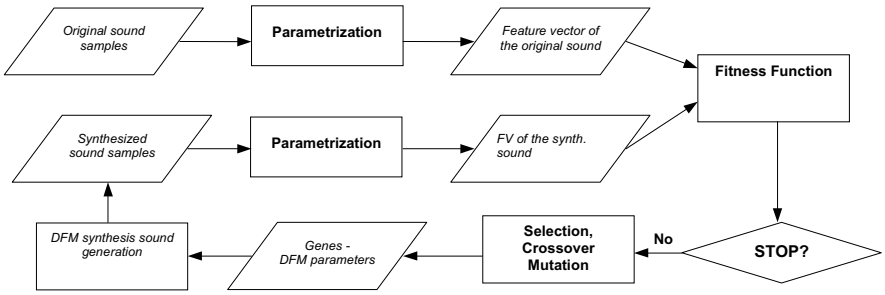


Fig. 3.49. Program flow chart

The starting point in this case is parameterization. Sound samples being identified are subject to automatic parameterization; the result is a feature vector of representative sound properties. Such a vector is then transmitted to the genetic algorithm input. The genetic algorithm forms chromosome populations that contain binary chains. The numeric presentation of such a chain is equivalent to the four parameters of the DFM synthesis - I_1, I_2, f_1, f_2 (Lim and Tan 1999). In the first place sound is synthesized for each chromosome according to the synthesis parameters I_1, I_2, f_1, f_2, coded in it. Then its automatic parameterization is carried out which is identical with the parameterization of the sound being identified. Such parameters of the synthetic sound are then compared with the corresponding model parameters of the sound being identified. A chromosome survives that transfers information about the synthesis of the sound that, with respect to sound parameters and according to the fitness function, is most similar to the modeled one. Other chromosomes are subject to mutation and crossing. The process is repeated for some consecutive chromosome generations. Finally, the most adapted chromosome that stores the parameters I_1, I_2, f_1, f_2 of the DFM synthesis are obtained. Such parameters are treated as features representing the instrument sound being identified and then transmitted to the input of the neural network. Information about the instrument class is additionally transmitted during the training phase of the neural network.

Thus, the decision making process proceeds in three main phases:

- Samples of the sound being analyzed are subject to automatic parameterization.
- The sound with parameter values most closed to those of the sound being analyzed are generated by means of the DFM synthesizer.
- The decision about the sound membership/assignment to a given class is taken based on the synthesis parameters.

Experiments carried out using the system are supposed to answer the question whether using sound synthesis parameters, obtained from the genetic algorithm processing, as the input vector for the neural network is the solution to sound identification process. The experiments shown above were partially conducted within the M.Sc. thesis of Leszczyna supervised by the author (Leszczyna 2002).

The effectiveness of the system was investigated based on a set of samples of musical instruments contained in the Catalogue of Musical Instrument Sounds of the Multimedia Systems Department, GUT (Kostek 1999). The program detects the beginning of the sound in the sample, performs the identification of the sound envelope phases (Attack and Sustain phase) and carries out the parametrization determining the number of available harmonics, brightness, even and odd harmonics contents, the first to the second harmonic amplitude ratio – for the Sustain phase, first, second and third Tristimulus – both for the Sustain and Attack phase – and the duration of the Attack phase. The pitch detection algorithm was based on Schroeder's histogram.

In one of the first steps of experiments a calibration of parameters of the genetic algorithm was performed in order to answer how numerous populations must be and how many generations are needed to achieve the preset degree of similarity. After the calibration of the genetic algorithm is done, the sound parameterization is carried out. As the result of this parameterization, four-dimensional vectors are obtained that include the parameters of the DFM synthesis and become the basis for investigations using neural networks. The experiments show, if vectors constructed this way are characteristic for instrument classes.

Genetic Algorithm

The aim of the implemented genetic algorithm is generating such four DFM synthesis parameters for which a synthesized sound is as similar to the original sound as possible, and this is according to the fitness function. Such parameters, together with determining the instrument class, form the basis of the training set for the neural network.

The DFM (*Double Frequency Modulation*) synthesis consists in signal generating, according to the following formula:

$$x[i] = A\sin(I_1 \sin(2\pi \frac{f_1}{f_s} i) + I_2 \sin(2\pi \frac{f_2}{f_s} i)) \tag{3.133}$$

where:

- $x[i]$ - i-th sample of synthesized signal,

- i - sample number,
- A - amplitude of synthesized signal,
- I_1, I_2 - modulation indices of two frequencies,
- f_1, f_2 – respective frequencies,
- f_S - sampling frequency.

When generating a signal, the parameters I_1, I_2, f_1, f_2 play the most important role, whereby their values decide on the timbre of the generated sound (Lim and Tan 1999). The higher the modulation index, the more widespread the signal spectrum. Although, some optimization procedures exist that select synthesis parameters which allow to achieve the desirable spectrum of a synthetic signal, the appropriate parameter selection is still a great challenge. In practice, trial-and-error methods or complex numerical methods are applied, in order to achieve this aim. Recently genetic algorithms were used for this purpose (Lim and Tan 1999). The relevant parameters which determine the spectrum of the synthesized waveform, can be evaluated by the fitness parameter. The smaller the fitness for a given set of parameters, the closer is the resultant spectrum to the desired one.

After the first chromosome population is generated, the estimation of chromosome adapting in the population is carried out at each step of the algorithm. Then the chromosome selection takes place and crossing and mutation are carried out. If the stop condition is not fulfilled, a new population is created. The selection of the first chromosome population is carried out randomly. The estimation of the chromosome adapting consists in the translation of a chromosome binary chain into the four DFM synthesis parameters and then – using them – in sound generating. Such a sound is then parameterized. The parameter similarity for both sounds is estimated according to the following formula:

$$\begin{aligned} z = {} & w_1 \times \left|P_O - P_S\right| + w_2 \times \left|T_{1O} - T_{1S}\right| + w_3 \times \left|T_{2O} - T_{2S}\right| + \\ & + w_4 \times \left|N_O - N_S\right| + w_5 \times \left|Br_O - Br_S\right| + w_6 \times \left|h_{evO} - h_{evS}\right| + \\ & w_7 \times \left|h_{oddO} - h_{oddS}\right| + w_8 \times \left|A_1 A_2 \operatorname{Re} l_O - A_1 A_2 \operatorname{Re} l_S\right| \end{aligned} \tag{3.134}$$

where:

- P_O, P_S – original and synthesized sound fundamental frequencies,
- T_{1O}, T_{1S} - *Tristimulus* I of the original and synthetic sound,
- T_{2O}, T_{2S} - *Tristimulus* II of the original and synthetic sound,
- N_O, N_S – the number of harmonics of the original and synthetic sound,
- Br_O, Br_S - original and synthetic sound brightness,

- h_{evO}, h_{evS} – number of even harmonics of the original and synthetic sound,
- h_{oddO}, h_{oddS} - the number of odd harmonics of the original and synthetic sound,
- $A_1A_2Rel_O$, $A_1A_2Rel_S$ - the amplitude ratio for the first and the second harmonics of the original and synthetic sound,
- w_i – weight coefficients, $i \in \langle 1,8 \rangle$.

It should be remembered that the lower the fitness function value, the more similar the sounds are.

After the estimation of the chromosome adapting in the whole population is done, 20% of the population remains unchanged and the other 80% is subject to random crossing and mutation. The roulette method is applied to select chromosomes to cross. However, mutation takes place according to uniform distribution.

In the program of musical instrument sound automatic identification, besides the fitness function weights, it is possible to determine the iteration number and the individual number in the population. In the parameter window of the DFM synthesis there are also the I_{max}, I_{min} and f_{max}, f_{min} values to be determined, as those values decide on the value ranges being generated. The narrower the range, the higher the effectiveness of the genetic algorithm operation. The genetic algorithm terminates after the preset iteration number is reached.

Neural Network

The aim of the implemented and trained neural network was the separation of training objects belonging to different instrument classes. A three-layer neural network was implemented, and the unipolar activation function was used.

In the program of musical instrument sound automatic identification the values of parameter I are determined directly based on values $I_{\max}$ and $f_{\max}$ according to the formula (Lim and Tan 1999):

$$\begin{aligned} n &= 2 \cdot ceil(\log_2(I_{\max})) + 2 \cdot ceil(\log_2(f_{\max})) \\ I &= n + 1 \end{aligned} \tag{3.135}$$

where n is the dimension of the network input vector, $ceil(x)$ denotes the function which returns the smallest integer that is greater or equal to x.

The formula (3.135) returns the minimum number of bits needed to record the four DFM synthesis parameters increased by 1. The additional (n+1)th node gets the constant value of 0 on its input. This node ensures

that the component w_{n+1} concerning the total stimulation of neurons is obtained (Hong and Chen 1996).

Experiments

The experiments using genetic algorithm consisted in carrying out an automatic parameterization of the set of musical instrument sound samples and in generating synthetic sound as similar to the original one as possible. Sound brightness was assumed to be similarity fitness. The fitness function had assumed the following form:

$$z = \left| Br_O - Br_S \right| \tag{3.136}$$

because the weights w_i, where $i \in \langle 1,4 \rangle \cup \langle 6,8 \rangle$, were assigned the value of 0 and weight w_5 was assigned the value of 1.

The first series of experiments consisted in a proper selection of two parameters of the genetic algorithm, i.e. the number of functional iteration and the number of individuals in the population. The selection of the DFM synthesis parameters, I_{max} and f_{max}, was also carried out. As the result of the investigations I_{max} was assigned the value of 0, and f_{max} 3200 Hz. It was assumed that the value of f_{max} mentioned above results from the analysis of the fundamental frequency of sounds that form the parametrization basis. The highest pitch is G7 which corresponds with the frequency of 3135 Hz. As already mentioned for the 10th iteration and the chromosome population size of 20, the maximum brightness distance between the pattern and the synthetic sound reached the value of 0.637. It had been assumed that doubled number of iteration (equal to 20) and the number of individuals in population equal to 20 are the values that make up a good compromise between the effectiveness of approximation and the time-consumption.

The parametrization of the set of 1167 sounds of 16 musical instruments, belonging to various groups, and played with differentiated articulation, was carried out. Afterwards, the statistical analysis of obtained parameter vectors $[I_1\, f_1\, I_2\, f_2]$ was performed. The average distance between the brightness of the original and the synthesized sound was equal to 0.138. Approximately 80% of all generated sounds differ from the original ones with respect to the average value of brightness. The difference between the brightness of natural and synthesized sounds is lower than 0.3 for 93% of sounds. In addition, values of Fisher statistics were calculated and the distribution over pairs of parameter values in a two-dimensional space was estimated. The analyses have shown that parameters are characterized by very low separability. These observations have proved that ac-

cording to a chosen fitness function the DFM synthesis parameters obtained in a non-deterministic process are not suitable for the use as elements of a vector of sound representative properties. Better separability of sounds was obtained only after adding back all other parameters to the fitness function, and in addition by manipulating weights of this function.

3.7 Other Technique-Based Classifiers

3.7.1 Decision Trees

Decision tree learning, referred to as a method for approximating discrete-valued target functions, is one of the most widely used and practical methods for inductive inference. Each leaf-node of a decision tree represents a complete classification of a given object, and each non-leaf node represents an attribute test. An attribute describes a characteristic of an object. Therefore, objects can be classified based on the results of successive attribute tests. Decision tree learning algorithms use a set of decision trees to represent a hypothesis space. Given a set of objects, the algorithm searches through this space to create a decision tree that most adequately partitions objects to different classes. An object is represented as a conjunction of variable values. Each variable has its own domain of possible values, typically discrete or continuous. The final decision tree returned by the algorithm represents the final hypothesis. Ideally, this hypothesis will correctly categorize future objects. A decision tree with a range of discrete class labels is called a classification tree, whereas a decision tree with a range of continuous values is called a regression tree. The space of all possible instances is defined by a set of possible objects that one could generate using these variables and their possible values. The well known programs for constructing decision trees are ID3 (Quinlan 1986, 1987), and CART (Classification and Regression Tree) (Breiman et al 1984). ID3 stands for "Iterative Dichotomizer (version) 3", later versions include C4.5 and C5 programs (Quinlan 1993);

A decision tree takes an object described by a set of attributes as an input, and at the output a yes/no decision is reached, thus representing Boolean functions. Decision tree learning is generally best suited to problems with the following characteristics
(http://www.cis.temple.edu/~ingargio/cis587/readings/id3-c45.html):

- Instances are represented by attribute-value pairs.

- Instances are described by a fixed set of attributes and their values.

Such algorithms assign classification to each example. The simplest case is when there are only two possible classes (Boolean classification). A more substantial extension allows learning target functions with real-valued outputs. Decision tree learning methods are quite robust to errors - both in classifications of the training examples and in the attribute values that describe these examples. The training data may also contain missing attribute values.

In general, decision trees represent a disjunction of conjunctions of constraints on the attribute-values of instances. Each path from the tree root to a leaf corresponds to a conjunction of attribute tests, and the tree itself corresponds to a disjunction of these conjunctions. More specifically, decision trees classify instances by sorting them down the tree from the root node to a leaf node, which provides the classification of the instance. Each node in the tree specifies a test of some attribute of the instance, and each branch descending from that node corresponds to one of the possible values for this attribute (see Fig. 3.50). An instance is classified by starting at the root node of the decision tree, testing the attribute specified by this node, and then moving down the tree branch corresponding to the value of the attribute. This process is then repeated at the node on this branch until a leaf node is reached (Quinlan 1993).

Since the resulting model is presented in the form of a tree structure, this visual presentation makes the decision tree model very easy to understand. As a result, the decision tree has become a very popular data mining technique. Decision trees are most commonly used for classification (http://www.cis.temple.edu/~ingargio/cis587/readings/id3-c45.html).

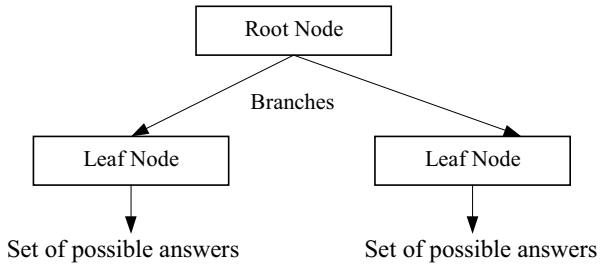


Fig. 3.50. Decision tree organization

Decision trees are applicable to the music recognition domain. It is worth seeing a paper by Herrera et al. in which a thorough review on classifiers can be found (Herrera et al 2000, 2003). Binary trees can be applied

to real-valued features. Such an approach can be found in Wieczorkowska's Ph.D. work and her later publications (Wieczorkowska 1999b; Wieczorkowska and Czyzewski 2003; Wieczorkowska et al 2003). She used a C4.5 algorithm for a classification of isolated musical sounds. In her studies a comparison between a rough set-based and a decision tree classifier accuracy while automatically identifying sounds was done. Typically, the decision tree classifier outperformed the rough set-based one, however, in the latter case a simple discretization method was used, such as Equal Interval Width Method. Another example of a decision tree application to musical sound classification is given by Jensen (Jensen and Arnspang 1999). During the learning stage, while descending the constructed decision tree, some questions are asked at each step of the analysis, then data are split into two groups, goodness of split (average entropy) is calculated and finally the question that renders the best goodness is chosen. Also, Foote applied a binary decision tree for audio classification in the context of content-based retrieval of music and audio (Foote 1997). He used the decision tree as a quantizer (Q-tree) of the parameterized music data. The algorithm automatically partitions the parameter space into quasi-separate classes. This process is supervised in the sense that each training example is associated with a class. The Q-tree is then used as a classifier (Foote 1997). The feature vectors consisted of 12 mel-cepstral coefficients (MFCC) and one additional parameter denoted as energy (Foote 1997).

3.7.2 Hybrid Analysis

Hybrid analysis offers an attractive paradigm for the design of intelligent systems for a broad range of applications. These applications refer to areas where for example such system features as robustness in the presence of noise, or modification of computational structures are needed.

Theoretical foundations of hybrid analysis include the integration of two, or more techniques. In literature there is a growing number of studies on interconnections between various realms of science, such as for example: neural networks and evolutionary computation, neural networks and rough sets, rough sets and areas such as knowledge discovery and data mining, intelligent information systems, multi-agent systems. Other combination of techniques, which are listed below are also encountered: rough sets are linked with decision system modeling and the analysis of complex systems, fuzzy sets, neural networks, evolutionary computing, data mining and knowledge discovery, pattern recognition, machine learning, and approximate reasoning. In particular, rough sets are used in probabilistic rea-

soning, granular computing (including information granule calculi based on rough mereology), intelligent control, intelligent agent modeling, identification of autonomous systems, and process specification.

Despite much research activity in the area of neural networks which has led to the discovery of several significant theoretical and empirical results and the development of important practical applications over the past decades, the design of artificial neural networks (ANN) for specific applications, under given sets of design constraints is, to a large extent, a process of trial and error, relying mostly on past experience with similar applications. Furthermore, the performance of ANN on particular problems is critically dependent on the choice of network architecture and the learning algorithms. ANNs essentially search for a suitable setting of weights within an otherwise a-priori specified network topology under the guidance from training samples. In order for this approach to succeed, the desired setting of parameters must in fact exist within the space being searched. Even when a suitable setting of parameters can be found using such an approach, the ability of the resulting network to generalize on data not seen during learning, the cost of the hardware realization, or the cost of using the network may be far from optimal. These factors make the process of ANN design difficult.

Thus, techniques for automating the design of neural architectures for particular classes of problems under a wide variety of design and performance constraints are clearly of interest. Motivated by this, some researchers have recently begun to investigate constructive or generative neural network learning algorithms that extend the search for the desired input-output mapping to the space of appropriately constrained network topologies by incrementally constructing the required network. Evolutionary algorithms (Holland 1975, Fogel et al 1966, Goldberg 1989, Koza 1992, Michalewicz 1992) offer an attractive and relatively efficient, randomized opportunistic approach to search for near-optimal solutions in a variety of problem domains. The design of efficient neural architectures for specific classes of problems under given sets of design and performance constraints is therefore a natural candidate for the application of evolutionary algorithms.

Generally, a genotype encodes a set of phenotypes or candidate solutions in the domain of interest, for example a class of neural architectures. Such encoding might employ genes that take on numeric values for a few parameters or complex symbol structures that are transformable into phenotypes (in this case, neural networks) by appropriate decoding processes.

A popular approach is the integration of neural network and fuzzy system to create a hybrid structure called neural fuzzy network (Ali and

Kamoun 1993; von Altrock 1995; Kosko 1992). Lin and Lee (1996) introduced the fuzzy adaptive learning control network (Falcon) to study hybrid structure parameter learning strategies, other authors used such techniques for classification purposes. Neural fuzzy (or neuro-fuzzy) networks such as the Generic Selforganizing Fuzzy Neural Network (GenSoFNN) (Tung and Quek 2002), Pseudo Outer Product Fuzzy Neural Network (POPFNN) (Quek and Zhou 1996), Adaptive Neuro Fuzzy Inference System (ANFIS) (Jang 1993), and Falcon (Lin 1996, Lin and Lee 1996), ART (Frank 1998; Lin and Lin 1996, 1997) are the realizations of the functionality of fuzzy systems using neural techniques. On the other hand, Ang and Quek (2004) proposed RSPOP: Rough Set-Based Pseudo Outer-Product Fuzzy Rule Identification Algorithm. The indicated that the pseudo outer-product (POP) rule identification algorithm used in the family of pseudo outer-product-based fuzzy neural networks (POPFNN) suffered from an exponential increase in the number of identified fuzzy rules and computational complexity arising from high-dimensional data. This decreases the interpretability of the POPFNN in linguistic fuzzy modeling. Their proposal concerns a novel rough set–based pseudo outer-product (RSPOP) algorithm that integrates the sound concept of knowledge reduction from rough set theory with the POP algorithm. The proposed algorithm not only performs feature selection through the reduction of attributes but also extends the reduction to rules without redundant attributes.

The enumerated applications of artificial neural networks to various fields drove development in theory. Due to this fact, some new trends in this domain appeared. One of these trends involves the compound structure of neural networks, so-called hierarchical neural networks. The basic network structure is composed of numbers of subnetworks. These subnetworks have a common input layer. Their middle layers are independent of one another. Every subnetwork has an assigned output node (Liqing 1998). Another trend that differs much from the all-class-one-network is the modular neural network concept. In this case, information supplied by the outputs of subnetworks can be fused by applying either the fuzzy or rough set approach. Hybrid methods have been developed by integrating the merits of various paradigms to solve problems more efficiently. It is often pointed out that hierarchical or modular neural networks are especially useful while discussing complex classification tasks involving a large number of similar classes. In such a case, one can refer to some sources that appeared recently in the literature (e.g. Auda and Kamel 1999, 2000; Chakraborty 2000, Mitra et al 1999; Pan et al 2000; Sarkar and Yegnanarayana 1999; Spaanenburg et al 2002; Szczuka 1998).

Feature subset selection by neuro-hybridization was presented as one of the most important aspects in machine learning and data mining applica-

tions by Chakraborty (2000). He engineered the neuro-rough hybrid algorithm that uses a rough set theory in the first stage to eliminate redundant features. Then, a neural network used in the second stage operates on a reduced feature set. On the other hand, Auda and Kamel (2000) proposed a modular neural network that consists of an unsupervised network to decompose the classification task across a number of neural subnetworks. Then, information from the outputs of such modules are integrated via a multimodule decision-making strategy that can classify a tested sample as 'vague class' or the boundary between two or more classes.

The paper by Peters et al. (2002; 2003) reviews the design and application of neural networks with two types of rough neurons: approximation neurons and decider neurons. The paper particularly considers the design of rough neural networks based on rough membership functions, the notion introduced by Pawlak and Skowron (1994). A so-called rough membership neural network consists of a layer of approximation neurons that construct rough sets. The output of each approximation neuron is computed with a rough membership function. Values produced by the layer of approximation neurons provide condition vectors. The output layer is built of a decider neuron that is stimulated by each new condition vector. A decider neuron compares the new condition vector with existing ones extracted from decision tables and returns the best fit. The decider neuron enforces rules extracted from decision tables. Information granules in the form of rules are extracted from decision tables using the rough set method (Pawlak and Skowron 1994). Other approaches based on modular and complex integral neural networks are also widely used in various problems as robust search methods, especially for uncertainty and redundancy in data (Komorowski et al 1998, 1999; Pal et al 2004; Polkowski et al 2002; Skowron et al 2000).

Examples of hybridization of intelligent computation techniques will be given further on in Chapters related to applications of cognitive processing of acoustical signals and subjective audio-visual correlation.

References

Agostini G, Longari M, Pollastri E (2001) Musical instrument timbres classification with spectral features. IEEE Multimedia Signal Processing

Ahn R, Holmes WH (1997) An improved harmonic-plus-noise decomposition method and its application in pitch determination. In: Proc IEEE Workshop on Speech Coding for Telecommunications, Pocono Manor, Pennsylvania, pp 41-42

d'Alessandro C, Yegnanarayana B, Darsinos V (1995) Decomposition of speech signals into deterministic and stochastic components. In: Proc IEEE International Conference on Acoustics, Speech, and Signal Processing, 1, pp 760-763

Ali M, Kamoun F (1993) Neural Networks for Shortest Path Computation and Routing in Computer Networks. IEEE Transactions on Neural Networks 4: 941-954

von Altrock C (1995) Fuzzy Logic & NeuroFuzzy Applications Explained. Prentice-Hall Intl Inc, New Jersey

Amerijck C, Verleysen M, Thissen P, Legat J (1998) Image Compression by Self-Organized Kohonen Map. IEEE Transactions on Neural Networks 9: 503-507

Ando S, Yamaguchi K (1993) Statistical Study of Spectral Parameters in Musical Instrument Tones. J Acoust Soc Am 94: 37-45

Ang KK and Quek C (2005) RSPOP: Rough Set–Based Pseudo Outer-Product Fuzzy Rule Identification Algorithm. Neural Comp 17: 205-243.

Auda G, Kamel M (1999). Modular Neural Networks: A Survey. International Journal of Neural Systems 9(2): 129-151

Auda G, Kamel M (2000) A Modular Neural Network for Vague Classification. Lecture notes in Computer Science; 2005, Lecture Notes in Artificial Intelligence, pp. 584-589

Auger F, Flandrin P (1995) Improving the Readibility of Time-Frequency and Time-Scale Representations by the Reassignment Method. IEEE Trans Signal Processing, 43, (5): 1068-1089

McAulay RJ, Quatieri TF (1986) Speech Analysis/Synthesis Based on a Sinusoidal Representation. IEEE Transactions Acoustic, Speech, Signal Processing ASSP-34:744-754

McAulay RJ, Quatieri TF (1990) Pitch estimation and voicing detection based on a sinusoidal speech model. In: Proc IEEE International Conference on Acoustics, Speech, and Signal Processing, 1, pp 249-252

Basseville M (1989) Distance Measures for Signal Processing and Pattern Recognition. Signal Processing, 18: 349–369

Bazan JG, Skowron A, Synak P (1994) Discovery of Decision Rules from Experimental Data. In: Lin TY, Wildberger AM (eds) Soft Computing Proc 3rd Intern Workshop on Rough Sets and Soft Computing. San Jose, pp 276-279

Bazan J, Szczuka M (2001) RSES and RSESlib - A Collection of Tools for Rough Set Computations. In: Proc of RSCTC'2000, LNAI 2005. Springer Verlag, Berlin

Bazan J, Szczuka M, Wroblewski J (2002) A New Version of Rough Set Exploration System. In: Alpigini JJ (ed) Proc RSCTC, LNAI 2475. Springer Verlag, Heidelberg, Berlin, pp 397-404

Bazan JG, Nguyen HS, Skowron A, Szczuka M (2003) A View on Rough Set Concept Approximations. In Wang G, Liu Q, Yao Y, Skowron A (eds) Proc of RSFD. Lecture Notes in Computer Science 2639. Springer, Chongqing, pp 181-188

Banzhaf W, Nordin P, Keller R, Francone F (1998). Genetic Programming – An Introduction, Morgan Kaufmann, San Francisco, CA

Box GEP (1957) Evolutionary operation: A method for increasing industrial productivity. J Royal Statistical Society 6(2): 81-101

Beauchamp JW (1993a) Detection of Musical Pitch from Recorded Solo Performances. In: Proc 94th Audio Eng Soc Conv, Berlin

Beauchamp JW (1993b) Unix Workstation Software for Analysis, Graphics, Modification, and Synthesis of Musical Sounds. In: 94th Audio Eng Soc Conv. Berlin, Preprint No 3479

Bershad N, Shynk J, Feintuch P (1993a) Statistical Analysis of the Single-layer Backpropagation Algorithm: Part I - Mean Weight Behaviour. IEEE Transactions on Acoustics, Speech, and Signal Processing 41: 573-582

Bershad N, Shynk J, Feintuch P (1993b) Statistical Analysis of the Single-layer Backpropagation Algorithm: Part II - MSE and Classification Performance. IEEE Transactions on Acoustics, Speech, and Signal Processing 41: 583-591

Bezdek JC, Hathaway RJ, Sabin MJ, Tucker WT (1987) Convergence Theory for Fuzzy c-Means Counterexamples and Repairs. IEEE Trans. Syst., Man, Cybern SMC-17, No 5

Billings SA, Chen S (1989) Extended Model Set, Global Data and Threshold Model Identification of Severely Non-Linear Systems. Int J Control 50: 1897-1923

Breiman, Friedman, Olshen, Stone (1984) Classification and Decision Trees Wadsworth

Brown JC (1992) Musical Fundamental Frequency Tracking Using a Pattern Recognition Method. J Acoust Soc Am 92: 1394-1402

Brown JC (1999) Computer identification of musical instruments using pattern recognition with cepstral coefficients as features. J Acoust Soc Am 105: 1933-1941

Carpenter G, Markuzon N (1998) ARTMAP-IC and Medical Diagnosis: Instance Counting and Inconsistent Cases. Neural Networks 11: 323-336

Carpenter GA, Grossberg S, Reynolds JH (1991) ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organising neural network. Neural Networks 4: 565-588

Cemgil AT, Gürgen F (1997) Classification of Musical Instrument Sounds using Neural Networks. In: Proc of SIU97

Chakraborty B (2000) Feature Subset Selection by Neuro-Rough Hybridization. In: Proc 2nd Int Conf on Rough Sets and Current Trends in Computing (RSCTC'2000), Banff, pp 481-487

Chang WC, Su AWY (2002) A novel recurrent network based pitch detection technique for quasi-periodic/pitch-varying signals. In: Proc IEEE, IJCNN, Hawaii, USA, pp 816-821

Chmielewski M, Grzymala-Busse J, Peterson N, Than S (1993a) The Rule Induction System LERS - a Version for Personal Computers. Foundations of Computing and Decision Sciences 18: 181-212

Chmielewski MR, Grzymala-Busse JW, et al (1993b) The Rule Induction System LERS - a Version for Personal Computers, Foundations of Computing and Decision Sciences 18, Poznan, pp 181-212

Chmielewski MR, Grzymala-Busse JW (1994) Global Discretization of Continuous Attributes as Preprocessing for Machine Learning. In: Lin TY, Wildberger AM (eds) Soft Computing, Proc 3rd Intern Workshop on Rough Sets and Soft Computing. San Jose, pp 294-301

Choi D, Park S (1994) Self-Creating and Organizing Neural Networks. IEEE Transactions on Neural Networks 5: 561-575

Choi H, Williams W (1989) Improved Time-Frequency Representation of Multi-Component Signals using Exponential Kernels. IEEE Transactions Acoustic, Speech, Signal Processing 37: 862-871

Christensen NS, Christensen KE, Worm H (1992) Classification of Music Using Neural Net, In: 92nd Audio Eng. Soc Conv, Preprint No 3296, Vienna.

Chui ChK, Montefusco L, Pucco L (eds) (1994) Wavelets - Theory, Algorithms and Applications. Academic Press Inc, San Diego

Cimikowski R, Shope P (1996) A Neural-Network Algorithm for a Graph Layout Problem. IEEE Transactions on Neural Networks 7: 341-345

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

Cosi P, De Poli G, Prandoni P (1994b) Timbre characterization with Mel-Cepstrum and Neural Nets. In: Proc of the 1994 ICMC, pp 42-45

Czyzewski A (1997) Learning Algorithms for Audio Signal Enhancement Part 2: Implementation of the Rough-Set Method for the Removal of Hiss. J of Audio Engineering Society 45, No 11: 931-943

Czyzewski A, Kaczmarek A (1993) Multilayer Knowledge Base System for Speaker Independent Recognition of Isolated Words. In: Proc RSKD-93, pp 411-420

Czyzewski A, Kaczmarek A (1994) Speech Recognition Systems Based on Rough Sets and Neural Networks. In: Lin TY, Wildberger AM (eds) Soft Computing, Proc 3rd Intern Workshop on Rough Sets and Soft Computing. San Jose, pp 97-100.

Czyzewski A, Kaczmarek A (1995) Speaker-independent Recognition of Isolated Words Using Rough Sets. In: Proc Joint Conf on Information Sciences. Wrightswille Beach, pp 397-400

Czyzewski A, Kostek B (1998) Tuning the Perceptual Noise Reduction Algorithm Using Rough Sets. Lecture Notes in Artificial Intelligence No 1424. In: Polkowski L, Skowron A (eds) Rough Sets and Current Trends in Computing. Proc RSCTC'98, Springer-Verlag, Heidelberg New York, pp 467-474

Czyzewski A, Krolikowski R (1998) Application of Fuzzy Logic and Rough Sets to Audio Signal Enhancement. In: Pal SK, Skowron A (eds) Rough-Fuzzy Hybridization: A New Trend in Decision-Making. Springer-Verlag, Singapore

Czyzewski A, Szczerba M, Kostek B (2002) Pitch Estimation Assisted by the Neural Network-Based Prediction Algorithm. In: Proc ISMA'2002, Mexico City, pp 246-255

Czyzewski A, Szczerba M, Kostek B (2004) Musical Phrase Representation and Recognition by Means of Neural Networks and Rough Sets. Rough Set Theory and Applications (RSTA) 1, pp 259-284, Advances in Rough Sets, Subseries of Springer-Verlag Lecture Notes in Computer Sciences, LNCS 3100, Transactions on Rough Sets, Grzymala-Busse JW, Kostek B, Swiniarski RW, Szczuka M (eds)

Dawis L (1991) Handbook of genetic algorithms, Van Nostrand Reinhold, New York

Dougherty J, Kohavi R, Sahami M (1995) Supervised and unsupervised discretization of continuous features. In: Proc 12th International Conference on Machine Learning, Los Altos, CA: Morgan Kaufmann, pp 194-202.

Downie JS (2003) Music information retrieval. In: Cronin B (ed) Annual Review of Information Science and Technology 37. Medrofd, NJ, Information Today, pp 295-340. Available from URL: http://music-ir.org/downie_mir_arist37.pdf

Dziubiński M, Kostek B (2004) High accuracy and octave error immune pitch detection algorithms. Archives of Acoustics 29: 3-23

Dziubinski M, Dalka P, Kostek B (2005) Estimation of Musical Sound Separation Algorithm Effectiveness Employing Neural Networks, J Intelligent Information Systems, Special Issue on Intelligent Multimedia Applications 24, No 2: 1333-157 (*in print*)

Elman J (1990) Finding Structure in Time. Cognitive Science 14: 179-211

Eronen A, Klapuri A (2000) Musical instrument recognition using cepstral coefficients and temporal features. In: Proc IEEE Intern Conference, ICASSP'2000

Eronen A (2001) Comparison of features for musical instrument recognition. In: IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA'01

Evangelista G (1993) Pitch-Synchronous Wavelet Representations of Speech and Music Signals. IEEE Transactions Signal Proc 41: 3313-3330

Fahlman S (1988) An Empirical Study of Learning Speed in Back-Propagation Networks. (Technical Report CMU-CS-88-162 of Carnegie Mellon University in Pittsburgh)

Fahlman S (1991) The Recurrent Cascade-Correlation Architecture. (Technical Report of Carnegie Mellon University in Pittsburgh)

Fahlman S, Lebiere S (1991) The Cascade-Correlation Learning Architecture. (Technical Report of Carnegie Mellon University in Pittsburgh)

Feiten B, Günzel S (1994) Automatic indexing of a sound database using self-organizing neural nets. J Computer Music 18: 53-65

Flangan J (1996) Self-Organisation in Kohonen's SOM. Neural Networks 9: 1185-1197

Flangan J (1997) Analysing a Self-Organising Algorithm. Neural Networks 10: 875-883

Fogel LJ, Owens AJ, Walsh MJ (1966) Artificial Intelligence Through Simulated Evolution. New York: Wiley Publishing

Fogel DB (1992) An analysis of evolutionary programming. In: Proc First Annual Conference on Evolutionary Programming, La Jolla, CA, pp 43-51

Foote JT (1997) A Similarity Measure for Automatic Audio Classification. In: Proc of the AAAI 1997 Spring Symposium on Intelligent Integration and Use of Text, Image, Video, and Audio Corpora. Stanford

Fragoulis DK, Avaritsiotis JN, Papaodysseus CN (1999) Timbre recognition of single notes using an ARTMAP neural network. In: Proc of the 6th IEEE International Conference on Electronics, Circuits and Systems. Paphos

Frank T, Kraiss K, Kuhlen T (1998) Comparative Analysis of Fuzzy ART and ART-2A Network Clustering Performance. IEEE Transactions on Neural Networks 9: 544-559

Fraser AS (1962) Simulation of Genetic System. J of Theoretical Biology 2: 329-346

Fraser AS, Burnell D (1970) Computer Models in Genetics. McGraw-Hill, New York

Fujinaga I (1998) Machine recognition of timbre using steady-state tone of acoustic musical instruments. In: Proc International Computer Music Conference, pp 207-10

Fujinaga I, McMillan K (2000) Realtime recognition of orchestral instruments. In: Proc International Computer Music Conference, pp 141-143

Fukushima K (1975) Cognitron: A Self-organizing Multilayered Neural Network. Biological Cybernetics 20: 121-136

Fukushima K (1988) Neocognitron: A Hierarchical Neural Network Capable of Visual Pattern Recognition. Neural Networks 1: 119-130

Fukushima K, Wake N (1991) Handwritten Alphanumeric Character Recognition by the Neocognitron. IEEE Transactions Neural Networks 2: 355-365

Genossar T, Porat M (1992) Can One Evaluate the Gabor Expansion Using Gabor's Iterative Algorithm. IEEE Transactions Signal Processing 40: 1852-1861

McGogenal CA, Rabiner LR, Rosenberg AE (1977) A subjective evaluation of pitch detection methods using LPC synthesized speech. IEEE Trans on Acoustics, Speech and Signal Processing, ASSP-25, (3)

Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley Publishing Company

Grey JM (1977) Multidimensional perceptual scaling of musical timbres. J Acoust Soc Am 61: 1270-1277

Grzymala-Busse DM, Grzymala-Busse JW (1994) Comparison of Machine Learning and Knowledge Against Methods of Rule Induction Based on Rough Sets, In: Rough Sets, Fuzzy Sets and Knowledge Discovery, Springer-Verlag, London

Grzymala-Busse JW, Kostek B, Swiniarski RW, Szczuka M (eds) (2004) Rough Set Theory and Applications (RSTA) 1, Advances in Rough Sets, Subseries

of Springer-Verlag Lecture Notes in Computer Sciences, LNCS 3100, Transactions on Rough Sets

Grzymala-Busse JW, Lakshmanan A (1992) LEM2 with Interval Extension: An Induction Algorithm for Numerical Attributes. In: Proc 4th International Workshop on Rough Sets, Fuzzy Sets, and Machine Discovery, RSFD-96. Tokyo, pp 67-73

Guillemain P, Kronland-Martinet R (1991) Parameters Estimation Through Continuous Wavelet Transform for Synthesis of Audio-Sounds. 90th Convention AES. Paris, Preprint No 3009 (A-2)

Herrera P, Amatriain X, Battle E, Serra X (2000) Towards Instrument Segmentation for Music Content Description: a Critical Review of Instrument Classification Techniques. In: Proc Intern Symposium on Music Information Retrieval, ISMIR 2000, URL: http://ismir2000.indiana.edu/

Herrera P, Peeters G, Dubnov S (2003) Automatic classification of musical instrument sounds. J New Music Research 32: 3-21

Hess W (1983) Pitch Determination of Speech Signals, Algorithms and Devices. Springer Verlag, Berlin Heidelberg

Hong T-P, Chen J-B (1996) Automatic Acquisition of Membership Functions by Data Analysis. In: Proc 4th International Workshop on Rough Sets, Fuzzy Sets, and Machine Discovery, RSFD-96. Tokyo, pp 315-319

Holland J H (1975) Adaptation in Natural and Artificial Systems, The University of Michigan

Horner A and Goldberg A (1991) Genetic Algorithms and Computer-Assisted Music Composition. In Proc of 1991 International Computer Music Conference, pp 479-482

Hunter J (2001) An overview of the MPEG-7 Description Definition Language (DDL). IEEE Transactions on Circuits and Systems for Video Technology 11: 765-772

Hu J, Xu S, Chen J (2001). A modified pitch detection algorithm. IEEE Communications Letters, 5, (2)

Ishikawa M (1996) Structural Learning with Forgetting. Neural Networks 9: 509-521

Ishikawa M (1997) Structural Learning and Rule Discovery. In: Proc 3rd Conf Neural Networks and Their Applications. Kule, pp 17-29

Janer L (1995) Modulated Gaussian Wavelet Transform based speech analyser pitch detection algorithm. In: Proc EUROSPEECH 1, pp 401-404

Jang J-SR (1993) ANFIS: Adaptive-Network-Based Fuzzy Inference Systems. IEEE Trans on Systems, Man, and Cybernetics 23, No 03: 665-685

Jelonek J, Lukasik E, Naganowski A, Slowinski R (2003) Inferring Decision Rules from Jurys' Ranking of Competing Violins. In: Proc Stockholm Music Acoustic Conference. KTH Stockholm, pp 75-78

Jelonek J, Lukasik E, Naganowski A, Slowinski R (2004) Inducing jury's preferences in terms of acoustic features of violin sounds. LNCS - Artificial Intelligence and Soft Computing 3070/2004, Springer Verlag, pp 492-497

Jensen K, Arnspang JJ (1999) Binary decission tree classification of musical sounds. In: Proc of the 1999 ICMC

Johnston J (1988) Transform Coding of Audio Signals Using Perceptual Noise Criteria. J IEEE Select Areas on Commun 6

De Jong KA (1992) Are genetic algorithms function optimizers? In: Proc Second International Conference on Parallel Problem Solving from Nature

De Jong KA and Spears W (1991) Learning concept classification rules using genetic algorithms. In: Proc Twelfth International Joint Conference on Artificial Intelligence, Sydney, Australia, pp 651-656

Kaczmarek A, Czyzewski A, Kostek B (1998) Investigating Polynomial Approximation for the Spectra of the Pipe Organ Sound. Archives of Acoustics 24: 3-24

Kaminskyj I (2000) Multi-feature Musical Instrument Sound Classifier. In: Proc Acust Comp Music Conf, Brisbane, pp 46-54

Kaminskyj I (2002) Multi-feature Musical Instrument Sound Classifier v/user determined generalisation performance. In: Proc ACMC 2002, Melbourne, pp 53-62

Kaminskyj I, Materka A (1995) Automatic source identification of monophonic musical instrument sounds. In: Proc of the IEEE International Conference On Neural Networks 1, pp 189-194

Kaminskyj I, Czaszejko T (2005) Automatic recognition of isolated monophonic musical instrument sounds using *k*NNC. J Intelligent Information Systems, Special Issue on Intelligent Multimedia Applications, 2005 (*in print*).

Kangas J, Kohonen T, Laaksonen K (1990) Variants of Self-Organizing Maps. IEEE Transactions Neural Networks 1, No 1: 93-99

Karnin Ed (1990) A simple procedure for pruning back-propagation trained neural networks. IEEE Transactions Neural Networks 1: 239-242

Kasi K, Zahorian SA (2002) Yet another algorithm for pitch tracking. In: Proc IEEE International Conference on Acoustics, Speech, and Signal Processing, 1, 361-364

Kay SM (1988) Modern Spectral Estimation: Theory and Application. Englewood Cliffs, New Jersey

Kim H-G, Berdahl E, Moreau N, Samour A, Sikora T (2003) Study of MPEG-7 sound classification and retrieval. ACM Multimedia, Berkley

Klapuri A (1999) Wide-band Pitch Estimation for Natural Sound Sources with Inharmonicities. In: Proc 106th Audio Eng Soc Conv, Munich, Preprint No 4906

Knecht W, Schenkel M, Moschytz G (1995) Neural Network Filters for Speech Enhancement. IEEE Trans. on Speech and Audio Processing 3, No 6: 433-438

Kohonen T (1990) The Self-Organizing Map. Proc of the IEEE 78: 1464-1477

Kohonen T, Kaski S, Lappalainen H (1997) Self-Organized Formation of Various Invariant-Feature Filters in the Adaptive-Subspace SOM. Neural Computation 9: 1321-1344

Kohonen T, Oja E, Simula O, Visa A, Kangas J (1996) Engineering Applications of the Self-organizing Map. Proc of the IEEE 84, No 10: 1358-383

Komorowski J, Pawlak Z, Polkowski L, Skowron A (1998) Rough Sets: A Tutorial. In: Pal SK, Skowron A (eds) Rough Fuzzy Hybridization: A New Trend in Decision-Making. Springer-Verlag

Komorowski J, Polkowski L, Skowron A (1999) Rough Sets: A Tutorial. In: Rough Fuzzy Hybridization: A New Trend in Decision-Making, Pal SK, Skowron A (eds.), Springer-Verlag, Singapore, 3-98.

Kosko B (1992) Neural Networks and Fuzzy Systems. Prentice-Hall Intl Inc, New Jersey

Kosko B (1997) Fuzzy Engineering. Prentice-Hall Intern Ed, New Jersey

Kostek B (1994) Application des réseaux de neurones pour l'analyse de l'articulation musicale. J de Physique IV 4: 597-600

Kostek B (1995a) Statistical Versus Artificial Intelligence Based Processing of Subjective Test Results. In: 98th Audio Eng Soc Conv, Preprint No. 4018, J Audio Eng Soc (Abstracts) 43, No 5: 403, Paris

Kostek B (1995b) Methoden kuenstlicher Intelligenz in Analysen des Musikklangs. In: Proc DAGA'95, Saarbruecken

Kostek B (1995c) Feature Extraction Methods for the Intelligent Processing of Musical Signals. In: Proc 99th Convention AES, Preprint No 4076 (H4), J Audio Eng Soc (Abstracts) 43, No 12, New York

Kostek B (1996) Rough Set and Fuzzy Set Methods Applied to Acoustical Analyses. J Intell Automation and Soft Computing – Autosoft 2: 147-158

Kostek B (1997) Sound Quality Assessment Based on the Rough Set Classifier. In: Proc EUFIT'97, Aachen, pp 193-195

Kostek B (1998a) Soft Computing-Based Recognition of Musical Sounds. In: Polkowski L, Skowron A (eds) Rough Sets in Data Mining and Knowledge Discovery I/II. Physica-Verlag (Springer-Verlag), pp 193-213

Kostek B (1998b) Soft Set Approach to the Subjective Assessment of Sound Quality, FUZZ-IEEE'98 (World Congress on Computational Intelligence), Anchorage, Alaska, USA, pp 669-674

Kostek B (1998c) Computer-Based Recognition of Musical Phrases Using the Rough-Set Approach. J Information Sciences 104: 15-30

Kostek B (1999) Soft Computing in Acoustics, Applications of Neural Networks, Fuzzy Logic and Rough Sets to Musical Acoustics. Physica Verlag, Heidelberg New York

Kostek B (2003) "Computing with words" Concept Applied to Musical Information Retrieval. Electronic Notes in Theoretical Computer Science 82, No 4

Kostek B (2004a), Application of soft computing to automatic music information retrieval. J American Society for Information Science and Technology 55, No 12: 1108-1116

Kostek B (2004b) Musical Instrument Classification and Duet Analysis Employing Music Information Retrieval Techniques. Proc of the IEEE 92, No 4: 712-729

Kostek B, Czyzewski A (2000) Automatic Classification of Musical Sounds. In: Proc 108th Audio Eng Soc Conv, Paris, Preprint No 2198

Kostek B, Czyzewski A (2001a) Automatic Recognition of Musical Instrument Sounds - Further Developments. In: Proc 110th Audio Eng Soc Conv, Amsterdam

Kostek B, Czyzewski A (2001b) Representing Musical Instrument Sounds for their Automatic Classification. J Audio Eng Soc 49: 768 – 785

Kostek B, Czyzewski A (2004) Processing of Musical Metadata Employing Pawlak's Flow Graphs. In: Rough Set Theory and Applications (RSTA), Advances in Rough Sets, Subseries of Springer-Verlag Lecture Notes in Computer Sciences, LNCS 3100, Transactions on Rough Sets 1, Grzymala-Busse JW, Kostek B, Swiniarski RW, Szczuka M (eds), pp 285-305

Kostek B, Krolikowski R (1997) Application of Neural Networks to the Recognition of Musical Sounds. Archives of Acoustics, 22, No 1: 27-50

Kostek B, Szczuko P, Zwan P (2004) Processing of Musical Data Employing Rough Sets and Artificial Neural Networks. In: Proc RSCTC 2004, Uppsala, LNAI 3066. Springer Verlag, Berlin Heidelberg, pp 539-548

Kostek B, Szczuko P, Zwan P, Dalka P (2005) Processing of Musical Data Employing Rough Sets and Artificial Neural Networks, Transactions on Rough Sets, Springer Verlag, Berlin, Heidelberg, New York, (*in print*).

Kostek B, Wieczorkowska A (1996) Study of Parameter Relations in Musical Instrument Patterns. In: Proc 100th Convention AES, Preprint No 4173 (E-6), Copenhagen, J Audio Eng Soc (Abstracts) 44, No 7/8: 634

Kostek B, Wieczorkowska A (1997) Parametric Representation of Musical Sounds. Archives of Acoustics 22, No 1: 2-26

Kostek B, Zwan P (2001) Wavelet-based automatic recognition of musical instruments. 142nd Meeting of the Acoustical Soc of Amer, Fort Lauderdale, Florida, USA, No 5 110, 4pMU5, p 2754

Kostek B, Zwan P, Dziubinski M (2002) Statistical Analysis of Musical Sound Features Derived from Wavelet Representation. In: Proc 112 Audio Eng Soc Convention, Munich, Germany

Kostek B, Zwan P, Dziubinski M (2003) Musical Sound Parameters Revisited. In: Proc of the Stockholm Music Acoustics Conference, SMAC'03, Stockholm

Koza JR (1991) Evolving a computer program to generate random numbers using the genetic programming paradigm. In: Proc Fourth International Conference on Genetic Algorithms, La Jolla, CA, pp 37-44

Koza JR (1992) Genetic Programming, MIT Press, Cambridge, MA

Krimphoff J, McAdams S, Winsberg S (1994) Caracterisation du Timbre des Sons Complexes. II Analyses acoustiques et quantification psychophysique. J de Physique IV 4: 625-628

Kunieda N, Shimamura T, Suzuki J (1996) Robust Method of Measurement of Fundamental Frequency by ACOLS-Autocorrelation of Log Spectrum. In: Proc IEEE Int Conf on Acoustics, Speech and Signal Processing 1, Atlanta, pp 232-235

Kwong S, Gang W, Lee CH (1992) A pitch detection algorithm based on time-frequency analysis. In: Proc of Singapore ICCS/ISITA, pp 432-436

Lenarcik A., Piasta Z (1994) Deterministic Rough Classifiers, in Soft Computing Lin TY, Wildberger AM (eds) Proc 3rd Intern Workshop on Rough Sets and Soft Computing, San Jose, CA, USA, pp 434-441

Leonarties IJ, Billings SA (1985) Input-Output Parametric Models for Non-Linear Systems Part II: Stochastic Non-Linear Systems. Int J Control 41: 329-344

Leszczyna R (2002) Musical sound idenfification by means of DFM synthesis and genetic algorithms MSc Thesis, Multimedia Systems Department, Gdansk University of Technology (*in Polish*), Kostek B (supervisor)

Lim SM, Tan BTG (1999) Performance of the Genetic Annealing Algorithm in DFM Synthesis of Dynamic Musical Sound Samples. J Audio Eng Soc 47: 339

Lin C-J (1996) A Fuzzy Adaptive Learning Control Network with On-Line Structure And Parameter Learning. International J of Neural Systems 7, No 5: 569

Lin C-T, Lee CSG (1996) Neural Fuzzy Systems: A neural-fuzzy synergism to intelligent systems. New Jersey, Prentice-Hall

Lin C-T, Lin C-J (1996) A Neural Fuzzy System with Fuzzy Supervised Learning. IEEE Transactions on Systems, Man, and Cybernetics, PART B: Cybernetics 26, No 5

Lin C-J, Lin C-T (1997) An ART-Based Fuzzy Adaptive Learning Control Network. IEEE Transactions on Fuzzy Systems 5, No 4

Lindsay AT, Herre J (2001) MPEG-7 and MPEG-7 Audio – An Overview. J Audio Eng Soc 49: 589-594

Liqing Z (1998) A New Compound Structure of Hierarchical Neural Networks. In: Proc of IEEE World Congress on Computational Intelligence, ICEC98, Anchorage, pp 437-440

Lukasik E (2003a) AMATI-Multimedia Database of Violin Sounds. In: Proc Stockholm Music Acoustics Conference, KTH Stockholm, pp 79-82

Lukasik E (2003b) Timbre Dissimilarity of Violins: Specific Case of Musical Instruments Identification. Digital Media Processing for Multimedia Interactive Services, World Scientific, Singapore, pp 324-327

Lukasik E, Susmaga R (2003) Unsupervised Machine Learning Methods in Timbral Violin Characteristics Visualization. In: Proc Stockholm Music Acoustics Conference, KTH Sztokholm, pp 83-86

Magoulas G, Vrahatis M, Androulakis G (1997) Effective Backpropagation Training with Variable Stepsize. Neural Networks 10: 69-82

Maher RC, Beauchamp JW (1994) Fundamental Frequency Estimation of Musical Signals using a two-way Mismatch Procedure. J of the Acoust Soc of Am 95(4): 2254-2263

Mallat S (1991) Zero-Crossings of a Wavelet Transform. IEEE Transactions on Information Theory 37, No 4: 1019-1033

Marple Jr SL (1987) Digital Spectral Analysis: with Applications. Englewood Cliffs, New Jersey

Martin KD (1999) Sound-Source Recognition: A Theory and Computational Model. Ph.D. thesis, MIT

Martin KD, Kim YE (1998) Musical instrument identification: A pattern-recognition approach. In: Proc 136th Meeting of the Acoust Soc of America, 2pMU9. Norfolk

Marques J, Almeida L (1986) A Background for Sinusoid Based Representation of Voiced Speech. In: Proc IEEE International Conference on Acoustics, Speech, and Signal Processing, Tokyo, pp 1233-1236

MATHEMATICA (1996) Wavelet Explorer. Wolfram Research, Champaign, Illinois

Medan Y, Yair E, Chazan D (1988) An accurate pitch detection algorithm, In: Proc 9th International Conference on Pattern Recognition, Rome, Italy, 1, pp 476-80

Medan Y, Yair E, Chazan D (1991) Super resolution pitch determination of speech signals. IEEE Trans on Acoustics, Speech and Signal Processing, ASSP-39: 40-48

Mei X, Pan J, Sun S (2001) Efficient algorithms for speech pitch estimation. In: Proc International Symposium on Intelligent Multimedia, Video and Speech Processing. Hong Kong, pp 421-424

Meyer Y (1992) Wavelets and Applications. Springer-Verlag, Paris

Michalewicz Z (1992) Genetic Algorithms + Data Structures = Evolution Programs

Mitra S, Pal KS, Banerjee M (1999) Rough Fuzzy Knowledge-based Network - A Soft Computing Approach. In: Pal SK, Skowron A (eds) New Trend in Decision-Making. Springer Verlag, Singapore Berlin Heidelberg, pp 428-454

Monro G (1995) Fractal Interpolation Waveforms. J Comp Music 19: 88-98

Morando M, Muselli M, Guariano M (1996) Musical Rhythm Recognition with Neural Networks. In: Proc IASTED, Artificial Intelligence, Expert Systems, and Neural Networks, pp 229-232, Honolulu, Hawaii, USA, 1996

Mourjopoulos J, Tsoukalas D (1991), Neural Network Mapping to Subjective Spectra of Music Sounds. In 90th Audio Eng Soc Conv, Preprint No 3064, Paris 1991, J Audio Eng Soc (Abstr) 39, No 5

Nguyen HS (1998) Discretization Problem for Rough Sets Methods, RSCTC'98, Warsaw. Lecture Notes in Artificial Intelligence, No 1424, Springer Verlag, Rough Sets and Current Trends in Computing. In: Polkowski L, Skowron A (eds), pp 545-552.

Nguyen HS, Nguyen SH (1998) Discretization methods in data mining. In: Rough Sets in Knowledge Discovery. Polkowski L, Skowron A (eds), Physica Verlag, Berlin 1998, pp 451-482.

Noll AM (1967) Cepstrum Pitch Determination. J Acoust Soc Am 14: 293-309

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

Orr RS (1993) The Order of Computation of Finite Discrete Gabor Transforms. IEEE Transactions Signal Processing 41, No 1: 122-130

Pal SK, Polkowski L, Skowron A (2004) Rough-Neural Computing. Techniques for Computing with Words. Springer Verlag, Berlin Heidelberg New York

Pan Y, Shi H, Li L (2000) The Behavior of the Complex Integral Neural Network. In: Proc 2nd Int Conf on Rough Sets and Current Trends in Computing (RSCTC' 2000). Banff, pp 585-592

Papadopoulos G, Wiggins G (1998) A Genetic Algorithm for the Generation of Jazz Melodies, Department of Artificial Intelligence University of Edinburgh

Pawlak Z (1982) Rough Sets, J Computer and Information Science 11: 341-356

Pawlak Z (1996) Data versus Logic - A Rough Set View. In: Proc 4th International Workshop on Rough Sets, Fuzzy Sets, and Machine Discovery (RSFD-96), Tokyo, pp 1-8

Pawlak Z (1998) Reasoning about Data – A Rough Set Perspective. In: Polkowski L, Skowron A (eds) Rough Sets and Current Trends in Computing, Lecture Notes in Artificial Intelligence No 1424, Proc RSCTC'98, Springer-Verlag, Heidelberg New York, pp 25-34

Pawlak Z (2003) Probability, Truth and Flow Graph. Electronic Notes in Theoretical Computer Science 82, International Workshop on Rough Sets in Knowledge Discovery and Soft Computing, Satellite event of ETAPS 2003, Elsevier, Warsaw

Pawlak Z (2004) Elementary Rough Set Granules: Towards a Rough Set Processor. In: Pal SK, Polkowski L, Skowron A (eds) Rough-Neural Computing. Techniques for Computing with Words. Springer Verlag, Berlin Heidelberg New York, pp 5-13

Pawlak Z, Skowron A (1994) Rough Membership Functions. In: Yager R, Fedrizzi M, Kacprzyk J (eds) Advances in the Dempster-Shafer Theory of Evidence. John Wiley & Sons, NewYork, pp 251-271

Peeters G., McAdams S, Herrera P (2000) Instrument Description in the Context of MPEG-7'. In: Proc of International Computer Music Conference, Berlin, Germany

Peters JF, Pawlak Z, Skowron A (2002) A Rough Set Approach to Measuring Information Granules. In: Proc 26th Annual International Computer Software and Applications Conference, Oxford, England

Peters JF, Skowron A, Synak R, Ramanna S (2003) Rough Sets and Information Granulation, Tenth International Fuzzy Systems Association world Congress IFSA, Bilgic T, Baets D, Kaynak O (eds.), Lecture Notes in Artificial Intelligence 2715 of Lecture Notes in Artificial Intelligence, Springer-Verlag, Istanbul, Turkey, pp 370-377

De Poli G, Prandoni P (1997) Sonological models for timbre characterization. J New Music Research 26: 170-197

De Poli G, Piccialli A, Roads C (1991) Representation of Musical Signals. MIT Press, London

Polkowski L, Pal SK, Skowron A (2002) Rough-Neuro-Computing: Techniques for Computing with Words. Springer-Verlag New York, Inc., NJ

Polkowski L, Skowron A (eds) (1998a) Rough Sets and Current Trends in Computing, Lecture Notes in Artificial Intelligence. Springer-Verlag, Heidelberg New York

Polkowski L, Skowron A (eds) (1998b) Rough Sets in Knowledge Discovery 1: Methodology and Applications. Physica-Verlag, Heidelberg New York

Polkowski L, Skowron A (eds) (1998c) Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems. Physica-Verlag (Springer-Verlag), Heidelberg New York

Pollard HF, Jansson EV (1982) A Tristimulus Method for the Specification of Musical Timbre. Acustica 51: 162-171

Press WH, Flannery BP, Teukolsky SA, Vetterling WT (1986) Numerical Recipes. University Press, Cambridge

Proakis JG, Manolakis DG (1999) Digital Signal Processing. Principles, Algorithms and Applications, 3rd edn. Prentice Hall International

Quek C, Zhou RW (1996) POPFNN: A Pseudo Outer-product Based Fuzzy Neural Network. Neural Netw 9(9): 1569-1581

Quian X, Kimaresan R (1996) A Variable Frame Pitch Estimator and Test Results. In: Proc IEEE Int Conf on Acoustics, Speech and Signal Processing 1, Atlanta, pp 228-231

Quinlan JR (1986) Induction of Decision Trees, Machine Learning, (1): 81-106

Quinlan JR (1987) Simplifying decision trees. International J of Man-Machine Studies 27: 221-234

Quinlan JR (1993) C4.5: Programs for Machine Learning Morgan Kauffman,

Rabiner L, Cheng MJ, Rosenberg AE, Gonegal C (1976) A Comparative Performance Study of Several Pitch Detection Algorithms. IEEE Transactions Acoustic, Speech, Signal Processing 24: 399-418

Rabiner LR, Schafer RW (1978) Digital Processing of Speech Signals. Englewood Cliffs, Prentice Hall

Rechenberg I (1973) Evolutionstrategie. Fromman-Holzboog, Stuttgart, Germany

Riedmiller M (1994) Advanced Supervised Learning in Multi-layered Perceptrons - from Backpropagation to Adaptive Learning Algorithm. Intl J of Computer Standards and Interfaces, Special Issue on Neural Networks 5

Riedmiller M, Braun H (1993) A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. In: Proc of the IEEE Intl Conference on Neural Networks, San Francisco, pp 586-591

Rife DC, Boorstyn RR (1974) Single-Tone Parameter Estimation from Discrete-Time Observations. IEEE Trans on Information Theory, IT-20(5): 591-598

Rife DC, Boorstyn RR (1976) Multiple Tone Parameter Estimation From Discrete Time Observations. Bell System Technical Journal 55: 1389-1410.

Rumelhart DE, Hinton GE, Williams RJ (1986) Learning internal representations by error propagation. In Rumelhart DE and McClelland JL (eds), Parallel Distributed Processing: Explorations in the microstructure of cognition 1: Foundations, Cambridge, Massachusetts, MIT Press.

Sarkar M, Yegnanarayana B (1999) Application of Fuzzy-Rough Sets in Modular Neural Networks. In: Pal SK, Skowron A (eds) New Trend in Decision-Making. Springer Verlag, Singapore Berlin Heidelberg, pp 410-427

Schroeder MR (1968) Period histogram and product spectrum: New methods for fundamental frequency measurement. J Acoust Soc Am 43: 829-834

Schroeder MR (1989) Self-Similarity and Fractals in Science and Art. J Audio Eng Soc 37

Schwefel H-P (1977) Numeriche Optimierung von Computer-Modelen Mittels der Evolutions-Strategie. Birkauser, Basel, Switzerland

Skowron A (1994a) Data Filtration: a Rough Set Approach. In Ziarko WP (ed) Rough Sets, Fuzzy Sets and Knowledge Discovery. Springer-Verlag, London, pp 108-118

Skowron A (1994b) Decision Rules Based on Discernibility Matrices and Decision Matrices. In: Lin TY, Wildberger AM (eds) Soft Computing, Proc. 3rd Intern. Workshop on Rough Sets and Soft Computing. San Jose, pp 6-9

Skowron A, Nguyen SH (1995) Quantization of Real Value Attributes: Rough Set and Boolean Reasoning Approach. (ICS Research Report 11/95, Warsaw)

Skowron A, Stepaniuk J, Peters JF (2000) Approximation of Information Granule Sets. In: Proc 2nd Int Conf on Rough Sets and Current Trends in Computing (RSCTC' 2000). Banff, pp 33-40

Slowinski R (1994) Rough Set Processing of Fuzzy Information. In: Soft Computing. Lin TY, Wildberger AM (eds). Proc. 3rd Intern. Workshop on Rough Sets and Soft Computing, San Jose, CA, USA, pp 142-145

Slowinski R, Stefanowski J, Susumaga R (1996) Rough Set Analysis of Attribute Dependencies in Technical Databases. In: Proc 4th International Workshop on Rough Sets, Fuzzy Sets, and Machine Discovery (RSFD-96), Tokyo, pp 284-291

Spaanenburg L, Slump C, Venema R, van der Zwaag B-J (2000) Preparing for Knowledge Extraction in Modular Neural Networks. In: Proc 3rd IEEE Signal Processing Symposium, Leuven, Belgium

Spears WM, and De Jong KA (1991) On the virtues of uniform crossover. In: Proc Fourth International Conference on Genetic Algorithms, pp 230-236, La Jolla, CA: Morgan Kaufmann

Srinivas M, Patnaik LM (1994) Genetic algorithms: A survey. IEEE Computer: 17-26

Stankovic L, Katkovnik V (1998) Algorithm for the Instantaneous Frequency Estimation Using Time-Frequency Distributions with Adaptive Window Width. IEEE Signal Processing Letters 5, (9): 224-227

Swiniarski R (2001) Rough sets methods in feature reduction and classification. Int J Applied Math Comp Sci 11: 565-582

Szczerba M, Czyzewski A (2005) Pitch Detection Enhancement Employing Music Prediction. J. Intelligent Information Systems, Special Issue on Intelligent Multimedia Applications 24, No 2: 223- 251 (*in print*)

Szczuka MS (1998) Rough Sets and Artificial Neural Networks. In: Pal SK, Skowron A, (eds) Rough Sets in Knowledge Discovery: Applications, Case Studies and Software Systems. Physica Verlag, Heidelberg New York, pp 449-470

Szczuko P, Dalka P, Dabrowski M, Kostek B (2004) MPEG-7-based Low-Level Descriptor Effectiveness in the Automatic Musical Sound Classification. In: Proc 116 Audio Eng Convention, Berlin, Preprint No 6105

Tadeusiewicz R (1988) Speech Signal (*in Polish*). WKiL, Warsaw

Talkin D (1995) A Robust Algorithm for Pitch Tracking (RAPT). Speech Coding and Synthesis. Elsevier, New York, pp 495-518

Toiviainen P, Tervaniemi M, Louhivuori J, Saher M, Huotilainen M, and Näätänen R (1998) Timbre Similarity: Convergence of Neural, Behavioral, and Computational Approaches. Music Perception 16: 223-241

Tsumoto S, Yao YY, Hadjimichael M (eds) (1998) Bulletin of International Rough Set Society 2, No 1

Tung WL and Quek C (2002) GenSoFNN: a generic self-organizing fuzzy neural network. IEEE Transactions on Neural Networks 13, No 5: 1075-1086.

Werbos P (1988) Backpropagation: Past and future. In: Proc of the IEEE International Conference on Neural Networks, pp 343-353

Wessel D (1979) Timbre space as a musical control structure. J Computer Music 3: 45-52

Wieczorkowska A (1999a) The recognition efficiency of musical instrument sounds depending, on parameterization and type of a classifier (*in Polish*). Ph.D. Thesis, Gdansk University of Technology

Wieczorkowska A (1999b) Wavelet based analysis and parameterization of musical instrument sounds. In: Proc of the International Symposium on Sound Engineering and Mastering ISSEM'99, Gdansk University of Technology, pp 219-224

Wieczorkowska A, Czyzewski A (2003) Rough Set Based Approach to Automatic Classification of Musical Instrument Sounds. Electronic Notes in Theoretical Computer Sciences 82

Wieczorkowska A, Wroblewski J, Slezak D, Synak P (2003) Application of temporal descriptors to musical instrument sound recognition. J Intell Inf Syst 21: 71-93

Wilson R, Calway AD, Pearson ERS (1992) A Generalized Wavelet Transform for Fourier Analysis, the Multiresolution Fourier Transform and its Application to Image and Audio Signal Analysis. IEEE Transactions on Information Theory 38, No 2: 674-690

Wize JD, Caprio JR, Parks TW (1976) Maximum-Likelihood Pitch Estimation. IEEE Transactions of Acoustic, Speech, Signal Processing 24: 418-423

Ying GS, Jamieson LH, Michell CD (1996) A Probabilistic Approach To AMDF Pitch Detection. In: Proc 1996 International Conference on Spoken Language Processing, Philadelphia, pp 1201-1204. URL: http://purcell.ecn.purdue.edu/~speechg

Zhang T, Kuo JC-C (1999) Heuristic approach for generic audio data segmentation and annotation. ACM Multimedia Conference, Orlando, pp 67-76

Zhang W, Xu G, Wang Y (2002) Pitch estimation based on circular AMDF. In: Proc IEEE International Conference on Acoustics, Speech, and Signal Processing, 1, pp 341-344

Ziarko W (1993) Analysis of Uncertain Information in the Framework of Variable Precision Rough Sets. Foundations of Computing and Decision Sciences 18, Poznan, pp 381-396

Ziarko W (ed) (1994) Rough Sets, Fuzzy Sets, and Knowledge Discovery. Springer- Verlag, London

Ziarko W (1996) Review of Basics of Rough Sets in the Context of Data Mining. In: Proc 4th International Workshop on Rough Sets, Fuzzy Sets, and Machine Discovery, RSFD-96. Tokyo, pp 447-457

Zurada J (1992) Introduction to Artificial Neural Systems. West Publishing Comp, St Paul

Zurada J, Malinowski A (1994) Multilayer Perceptron Networks: Selected Aspects of Training Optimization. Applied Mathematics and Comp Science 4, No 3: 281-307

Zwicker E, Zwicker T (1991) Audio Engineering and Psychoacoustics: Matching Signals to the Final Receiver, the Human Auditory System. J Audio Eng Soc 39: 115-126

URL: http://ismir2004.ismir.net/ Music Information Retrieval website (2004)

URL: http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm MPEG-7 standard inormation (2003)

URL: http://www.cis.temple.edu/~ingargio/cis587/readings/id3-c45.html information on ID3 and C4.5 decision tree algorithms (2004)

URL: http://www.cs.uwyo.edu/~wspears/overview/ (information on genetic algorithms (2005)

URL: http://logic.mimuw.edu.pl/~ rses/ RSES homepage (2004)

URL: http://www.soft-computing.de/def.html

URL: http://www.rsds.wsiz.rzeszow.pl/rsds.php

URL: http://www.w3.org/Metadata/ (2001) (information on metadata)

4 COGNITIVE APPROACH TO MUSICAL DATA ANALYSIS

4.1 Musical Signal Separation

Digital signal processing is one of the most rapidly developing areas of science. With the explosive expansion of the Internet, the number of very demanding computer network users increases. Content analysis and searching for specific content are relatively new areas, and therefore new concepts and algorithms of processing them appear quite often. Currently there are no faultless solutions. Sound data analysis is connected with difficulties in analytical description as well as with significant redundancy characterized by high entropy included in the very type of information. Such characteristics also prevail for sound separation problems, hence the number of algorithms for sound separation from musical material. In addition, one should notice that there are some limitations regarding percussion sounds and other non-harmonic sources. Easy extraction of such sounds by means of existing algorithms is not possible. Therefore one could state that such instruments are a source of noise for an algorithm, which makes its operation more difficult, and decreases the accuracy and credibility of the result. Concurrently, one should select the musical material for analysis based on the instrumentation, avoiding non-harmonic sounds. In addition, the articulation such as glissando or tremolo causes the problem of detecting fundamental frequency in the spectrum. Another important factor should also be considered: the music of Western culture is based on simple relations of frequency to fundamental frequency. Therefore it is an obvious consequence that harmonic tones overlap in the spectrum, which makes the operation of most algorithms more difficult. The most basic notion in musical acoustics i.e. sound timbre, as mentioned in Chapter 2, remains unresolved, even if many important research works have been done in the field (e.g. McAdams et al 1995; Cosi et al 1994; Grey 1997; Krimphoff et al 1994; De Poli and Prandoni 1997; Toiviainen et al 1998, Wessel 1979).

Bożena Kostek: *Perception-Based Data Processing in Acoustics*, Studies in Computational Intelligence (SCI) **3**, 187–313 (2005)
www.springerlink.com

Among main methods for musical signal separation three basic concepts are involved: blind separation based on the statistical analysis of the signal, analysis and resynthesis of the spectrum of instruments included in the mixed material, and the use of directional information in the data extraction process. In this chapter an algorithm within the second concept mentioned is discussed more thoroughly, since it offers significant effectiveness with easy adaptation to conditions in which separation is performed.

Walmsley points out that an important aspect of an analysis is the ability to segregate events hierarchically. At the lowest level a signal model may be useful, whereas at the highest one a musical model based on probabilistic, heuristic or learning rules to track or predict source activity could be helpful (Walmsley 2000).

Other methods should be used in the separation of singers' voices, since in this case spectrum often becomes non-harmonic, the articulation abilities are greater and the voices often sing the melody in unison. The starting point for those methods should be getting acquainted with the mathematical model of a voice tract. In further steps also the decision-making systems based on artificial neural networks are useful as is the parameterization based on the spectral and cepstral analysis. The conclusion is that for each type of musical material a different algorithm should probably be used. A duet, chamber orchestra recordings, a sound in an instrument playing chords or multiple sound voices require different techniques for their effective separation.

In the process of the algorithm quality verification, various techniques may be used, and especially valuable is subjective testing. However, in a case-study shown in this chapter artificial neural networks have been used, trained with a collection of selected sound data from various instruments. For such a type of application the best results are obtained through a supervised training. The objective is to assign automatically a given sound to an appropriate instrument class after the separation was performed.

4.1.1 Musicological and Psychoacoustical Overview

Sound separation from a polyphonic recording is performed in many different ways. The core operation is to extract instrument or voice tracks from the sound mixture with the highest possible quality. The term 'polyphony' may be discussed in a number of approaches in music theory. Its literal translation from Latin is 'many voices'. The existence of several independent melodic lines describes polyphony in a strictly theoretical understanding, which was established as a technique of composition in Middle Ages, and bloomed in Baroque. Its most important feature was the use

of counterpoint and parallel existence of several melodies. The most complex forms could include up to 32 voices at the same time. The plurality of voices can be achieved with just one or with a number of instruments. A different perspective on the notion of multi-voice performances comes from modern music performances, where generally the division of playing musicians involves the soloist playing the melodic line, and the accompaniment consisting of the bass line and the harmonic structure of the tune, aimed at the completion of a melody through chords. Therefore, it is not necessary that the melodies in the voices have the same character. Playing chords also has the character of polyphony. Similar situation can be observed in vocal performances, where several people can sing at the same time, with or without accompaniment. From the viewpoint of algorithm design, the ability to detect actions taking place in the vocal tract and cognitive factors allowing for the description of phenomena on a higher level of abstraction is important.

Perceptual Modeling

In the design of a separating algorithm the knowledge of a psychophysical element of sound phenomena is significant. So far there is no universal model describing the operations of the auditory system and the nervous system with a mathematical approach. Logarithmic sensitivity of hearing is the basis for the analysis of perception. The cochlea acts as a bank of independent band-pass filters located in appropriate critical bands. The knowledge concerning that property caused changes in the approach towards musical signal decomposition into frequency components. Algorithms operating through a bank of cochlear filters are now more often used in such applications. Musical signal processing may be preceded by a pattern processing stage reflecting the place-domain model of sound in the cochlea, or each channel of the cochlea may be preceded with the level of sound pitch detection acting according to time-domain rules. There are also hybrid models referring to both solutions.

Output signals from cochlear filters go through a detection threshold level simulating the operations of an auditory nerve. The level of amplification by signal compression, regulates its dynamics, while autocorrelation detects periodicity in each of the sub-bands, thus simulating responses of neurons, each induced according to the phase of the stimulating musical signal. The response of the autocorrelation system is often illustrated by a two-dimensional (time, frequency) or three-dimensional (time, frequency and autocorrelation lag) representation called correlogram.

Meddis and Hewitt proposed a more complex model, including external ear characteristics, and a more complex model of auditory nerve operations

(Meddis and Hewitt 1991). Karjalainen and Tolonen used a similar model for multi-sound estimation, employing a half-wave rectifier and a low-pass filter in the detection threshold level (Karjalainen and Tolonen 1999). Similar approach was undertaken by Ellis, who introduced a three-dimensional correlogram (Ellis 1996). The set of delaying lines with logarithmically presented delay times is used for the calculation of autocorrelation with a constant number of samples per octave. The representation of two variables: a low-pass frequency and the logarithmically presented delays is a set of a correlogram 'slices'. Each cell is the short-time autocorrelation of the envelope of one of the filterbank frequency channels. This approach analyzes dense, ambient sound examples as a part of a vocabulary of a computational auditory scene analysis. A discrete element describing individual periodic sources, along with the algorithm for extracting them from the correlogram-based representation, is called the weft.

Subsequent phases of information processing are based on the blackboard system, where several parallel processing tasks take place. In this case the conclusion-drawing strategy is based on prediction, for which specific expectations are generated from the source model. The predictions are then approved or rejected. The objective of such approach is the musical environment analysis with special focus on the sounds coming from the external environment, for example ambient sounds.

In his work, Mellinger used the cochleagram sound representation and image processing techniques, on the basis of which the algorithm was making appropriate decisions while identifying patterns such as vibrato, or note onsets (Mellinger 1991). A time-frequency processing kernel set was introduced and used for the extraction of specific features from the correlogram. The proposed algorithm improves the detection of such features as the beginning and/or the end of a sound, and the frequency change, which is related to the fact that with a logarithmic frequency scale all changes of a period of harmonic vibrations are subject to coherent changes in fixed intervals on the frequency axis. The Mellinger's model concerns a higher modeling level reflecting musical event detection through the grouping of harmonic components using the combination according to the psychological theory of Gestalt. Individual aliquots become a part of a group, which reflects a sound event. It consists of components which are closely related with each other through the similarity of appropriate features, for example mutual attachment time or the same periodicity (Mellinger 1991).

Slaney, Naar and Lyon describe the technique of obtaining the auditory model directly from the correlogram, whose components with common periodicity are separated and used for the calculations of a short-term power spectrum of the signal, which leads to the creation of a cochleagram (Slaney et al 1994). Component sounds are resynthesized using an overlap-

add method. The auditory model by Slaney et al. may be described as an approximately constant Q-filtering and autocorrelation estimation. Weintraub also computes the autocorrelation of signals from the cochlear filter bank outputs. This enables to find information on periodicity in each of the frequency regions, from which the spectral estimate of each sound is generated (Weintraub 1986). The iterative algorithm locally maximizes the probability of a spectral estimate, taking into consideration the information about local periodicity and spectral continuity constraints. The approach was created for voice separation algorithms and becomes useful also in separating non-harmonic sounds. However, the system using the above-mentioned model must be trained with an appropriate data set, because it uses Hidden Markov models (HMM) for voicing state detection. A model of the cochlea by Lyon and Mead's consists of a cascade of 88 biquadratic filters implemented by means of the FPGA technology. Based on the set of sample inputs, SNRs of every filter outputs are calculated, resulting also in a cochleagram.

Another interesting approach was introduced in the study of de Cheveigne (1993), in which a time-domain comb filter removes all harmonic components from the signal. To make it possible, one must properly detect the fundamental frequency, the estimate of which is made using the AMDF function (Average Magnitude Difference Function). The task requires finding the minimum over the lag-domain of the function (r). For two sounds sounding at the same time a comb filter is used with two lags, and the double difference function (DDF) must be searched for the minimum over two dimensions. De Cheveigne's method, known also as the neural comb filter, works for sounds, for which fundamental frequencies do not overlap, but it requires a clear musical material of good quality.

The models discussed in the following subsections concern the psychophysical approach to sound perception. Higher level auditory processes are not characteristic for the analysis of strictly musical data, but they offer universality in complex mixtures of speech, music and external environmental sounds. There are also a number of models created especially for strictly musical data.

Harmonic Overlapping

There are many algorithms used for the analysis of frequency, envelope or sinusoidal signal phases. The problem with the separation of a mixture of sounds results from the fact that harmonic elements of a periodic and quasi-periodic signal take up a very broad band of frequency and it often happens that the components of different instruments or even harmonic elements in a chord played on one instrument may be located very close to

each other, or even overlap. This causes two major problems (Klapuri 1998):

- harmonics of different sounds are usually located in the same frequency bands and difficulties in assigning harmonic components to appropriate sources appear,
- when two sinusoids overlap at the same frequency, it becomes impossible to regain information about their envelope and phases.

Two properties should then be analyzed (Klapuri 1998):

- When one harmonic component h_j^S of a signal S overlaps component h_i^R of the interfering source R, then the fundamental frequency of a signal R equals $f_{0R} - m/n \cdot f_{0S}$, where $m, n \in N$.
- When fundamental frequencies of two sounds S and R are respectively f_{0R} and $f_0R - m/n \cdot f_{0S}$, then each nth harmonic component h_{nk} of signal R overlaps the mth harmonic component h_{mk} of the source S (k=1,2,3,4…).

It might seem that the above presented conditions are only theoretical assumptions. However, they are connected with the rules, which are the basis of Western culture music, where fundamental frequencies of sounds often reflect the above mentioned relations.

Two notes are in a harmonic relation, when their harmonic components reflect the following rule:

$$f_{02} = \frac{m}{n} \cdot f_{01} \tag{4.1}$$

where m, n are small integers. The smaller are the values, the closer the harmonic relation between them. For example, frequencies $\frac{4}{4}f, \frac{5}{4}f, \frac{6}{4}f$ make a chord in major key, while $\frac{4}{6}f, \frac{4}{5}f, \frac{4}{4}f$ make a chord in minor key. Notes are positioned in a logarithmic scale on axis f, where k symbolizes the note number:

$$f_{0k} = 440 \cdot 2^{\frac{k}{12}} [\text{Hz}] \tag{4.2}$$

For example for the piano, the values of k fall in the range of k=-48 to k=39. The combination of values of coefficients m and n enable the creation of intervals. Their positioning for the just intonation system can be seen in Table 4.1. The last column presents tuning the intervals to the equal tempered scale.

Table 4.1. Comparison of sound frequencies in just and equal tempered scales

Note 1	Note 2	Equal Temperament $f_0(N_2)$: $f_0(N_1)$	m	n	Just Intonation m:n	Difference [%]
C#	C	1.0595	16	15	1.0667	-0.68
D	C	1.1225	9(8)	8(7)	1.125(1.143)	-0.23 (+1.8)
D#	C	1.1892	6	5	1.2	-0.91
E	C	1.2599	5	4	1.25	+0.79
F	C	1.3348	4	3	1.333	+0.11
F#	C	1.4142	7	5	1.4	+1.0
G	C	1.4983	3	2	1.5	-0.11
G#	C	1.5874	8	5	1.6	-0.79
A	C	1.6818	5	3	1.667	+0.91
A#	C	1.7818	16(7)	9(4)	1.778(1.75)	+0.23 (-1.8)
B	C	1.8877	15	8	1.875	+0.68

Today the instruments are tuned according to the equal tempered scale, in which intervals are built according to the logarithmic relations. This in itself relates to the property of hearing resulting from Weber–Fechner's law. Giving up the tuning according to quotient relations enables the movement of harmonic trajectories on the frequency scale, making it difficult for them to overlap. For human hearing the difference is practically unnoticeable, with the exception of the third and minor seventh intervals. Musical terminology describes the case of minor seventh as a so-called blue note if it is played in a manner based on frequency division, not according to logarithmic placement. It is especially noticeable in the case of African music, on the basis of which blues was created (with its characteristic use of blue note).

In practical algorithms frequency resolution is limited; it is too small to differentiate tempered sounds or sounds that are located very close to each other on the frequency scale. The following example presents the case of harmonic components overlapping in the case of a chord in C-major key. It consists of three sounds: C, E and G, constituting a prime, a major third, and a fifth. They are characterized with the division coefficient, respectively: 1, 5/4 and 3/2. Each $5 \cdot n$ harmonic component of sound C will overlap harmonic component $4 \cdot n$, where n is a natural number. Similarly for C and G, where each $5 \cdot n$th will overlap the $3 \cdot n$th component, so in the case of the fifth the situation is even worse. Similar phenomena take place between C and G: each $6 \cdot n$th harmonic third overlaps the $5 \cdot n$th harmonic fifth. The presented example is very simple, because major chords are among the simplest harmonic structures in music. In reality more complex harmonic arrangements are used – ones which include chords with transpositions and their inversions, as well as chords expanded with external

sounds. The more consonant the chord, the greater the probability of overlapping of harmonic components. It is unavoidable, because such chords constitute the concept of music shaped in the Western civilization. In the case of harmonic relations which are fractional numbers, various algorithms separating the energy of overlapping aliquots into a number of sounds are used. In the case of an octave and its multiplication the problem is practically unsolvable.

4.1.2 Signal Separation Algorithms

Today many domains use the computerized methods of information processing. Very often they have the character of musical data, with specific characteristics forcing the use of appropriate processing methods. As one could expect, they do not always produce satisfying results, because in most cases information about the method of mixing specific sources is not available. It also happens very seldom that each channel transfers sounds from just one instrument or voice. Besides, different numbers of recording microphones are used, with different positioning and different methods of combining the signals arriving from the acoustic environment. The character of audio material and the a priori knowledge of the recording process allow for the selection of an appropriate separation algorithm, because each of them has its own limitations. So far there is no uniform, optimum and universal software producing satisfying results in the analysis of different kinds of audio material. Currently, three main trends exist, on the basis of which various versions of algorithms for polyphonic recording separation are created:

- blind separation algorithms, using the a priori information about static features of sources,
- algorithms that make spectrum separation and then decomposition possible,
- algorithms using information of spatial positioning of sources.

Blind Separation Algorithms

The problem of blind separation is connected with separating tracks, when there are a number of signal sources and several microphones. At the same time the characteristic of the channel is unknown. The general scheme of the blind separation algorithm is illustrated in Fig. 4.1 (Chan 1997).

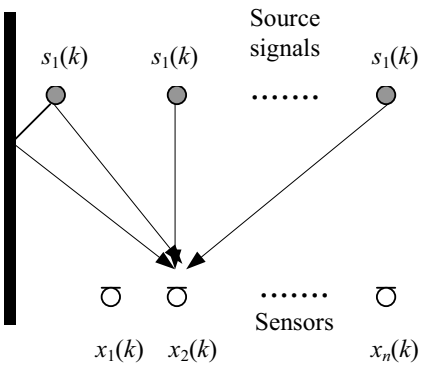


Fig. 4.1. Blind separation principle illustration (Chan 1997)

In recent years a number of new solutions developing the concept of blind separation surfaced – applications for hearing loss compensation (e.g. Greenberg et al 1992; Kates and Weiss 1996; Merks et al 1999), noise reduction (e.g. Lasecki and Czyzewski 1999), localization in multi-source environment, multimedia and teleconferencing (e.g. Hawley et al 1999; Kostek et al 1999; Mapp et al 1999), sound segregation (e.g. McAdams 1989), virtual auditory display (e.g. Bregman, 1990; Langendijk 2000), spatial filtration (e.g. Czyzewski 2003; Lasecki et al 1999), speech intelligibility improvement (e.g. Hawley et al 1999; Lasecki et al 1998) or even for use in passive sonars or image processing. They differ in the method of the concept implementation through the use of neural networks, genetic algorithms, higher level statistics, mutual information minimization, beamforming, or adaptive noise cancellation. Despite the difference in the ideas, the fundamental rule is that each source must be statistically independent (Amari et al 1996; Cardoso 1992; Choi et al 2001). Yellin and Weinstein (1994) proved that if sources are statistically independent, the necessary and sufficient separation condition is the statistic independence of system outputs.

Mutual independence of signal sources s_i is achieved when and only when the density of total probability $p_s(s)$ equals the product of boundary probabilities $p_{si}(s_i)$ (Torkkola 1999).

$$p_s(s) = \prod_{i=1}^{n} p_{s_i}(s_i) \tag{4.3}$$

Separated system outputs are independent in pairs, therefore they are mutually independent if outputs are linear functions of sources.

In this case the term 'blind' refers to the lack of the a priori knowledge about signals in the propagation channel from the sources to the recording system. In many practical situations the observations may be modeled using a linear mixture of input sources, i.e. a typical system of many inputs and many outputs. Defining the algorithm requires proper mathematical description. It is assumed that there are n observations of $x_1(k)$ ... $x_2(k)$, which are the mixture of n independent source signals $s_1(k)$ ... $s_2(k)$. The objective is to find n outputs of a system $y_1(k)$ ... $y_2(k)$, while $y_i(k) = z_j(s_j(k))$ i,j=1,…,n, where $z_j(*)$ is an unknown filtration operator. Thus $x(k) = [x_1(k), x_2(k), ..., x_n(k)]^T$ and similarly for $s(k)$ and $y(k)$. Mixing function $f(*)$ and separating $g(*)$ is calculated according to the following relations (Chan 1997):

$$X(k) = f(s(k)) \tag{4.4}$$

$$Y(k) = g(x(k)) \tag{4.5}$$

Thus $y(k) = g(f(s(k)))$

The objective is to find $g(*)$, for which:

$$\begin{pmatrix} y_1(k) \\ ... \\ y_n(k) \end{pmatrix} = P \begin{pmatrix} z_1(s_1(k)) \\ ... \\ z_n(s_n(k)) \end{pmatrix} \tag{4.6}$$

where P is the permutation matrix.

Conditions are simplified with the assumption that the mixing operation is linear and constant in time (*LTI*) and that it takes place without delays. Then $x(k) = As(k)$, where $\boldsymbol{A}$ is the matrix of constants of values $n \cdot n$. If it is reversible, then the resulting matrix $\boldsymbol{C} = \boldsymbol{A}^{-1}$. In such case the equality $y(k) = PCx(k)$ takes place for an ideal example. In practical applications a delay between matrix components exists, thus the adequate models are more complex (Balan et al 2001; Torkkola 1996, 1999; Westner and Bove 1999).

Independent Component Analysis (*ICA*)

The key element of *Independent Component Analysis* is the method of analysis of the main PCA (*Principal Component Analysis*) component, the objective of which is to achieve component vectors orthogonal to each other, whose linear combination will give possibly greatest variance (Abed-Meraim et al 1996; Amari 1999; Belouchrani et al, 1997; Liebermeister 2002). The vectors are calculated as n largest vectors of own covariance matrix, while the number of sources is n.

The ICA method is based on a number of basic limitations (Chan 1997; Jung 2001; Mansour et al 2000):

- the number of sources cannot be greater than the number of microphones,
- sources must be mutually independent statistically or must have possibly the smallest dependence level. The dependence cannot be measured due to the lack of signal power density information. For speech signals the independence of the sources of conditions is practically always met,
- input signals must be stationary with a zero mean value. Often a single unit variance for all sources is required (Jung and Kaiser 2003),
- only one of the sources can have Gaussian positioning (since a linear combination of Gauss processes also gives a Gauss process, which makes it impossible to differentiate between component processes),
- signals from sources must be synchronized,
- only the direction of vector **A** can be recreated in the separation process; amplification and sign cannot be recreated,
- processes should have a super-Gauss positioning, which is the key element in the selection of an appropriate algorithm. Signals with such positioning include music and speech.

The appropriate ICA algorithm usually uses slightly more advanced methods for demixing matrix estimation. A characteristic feature is the use of iterative methods for the statistical parameter optimization (Bell and Sejnowski 1995a, 1995b; Torkkola 1996, 1999). The transformation of the signal entropy equation gives:

$$H(y) = -\int p(x) \cdot \log[p(x)] \cdot dx = -E\{\log[p(x)]\} \tag{4.7}$$

$$H(x) = -\int p_i(x) \cdot \log[p_i(x)] \tag{4.8}$$

Total entropy is calculated using the following equation:

$$H(y_1, y_2, ..., y_N) = H(y_a) + H(y_2) + ... + H(y_N) - I(y_1, y_2, ..., y_N) \tag{4.9}$$

where $I(Y)$ is a mutual information.

If sources are independent from each other, then $I(y) = 0$. Otherwise the number will always be greater than zero. It should also be noted that minimization $I(Y)$ is connected with the maximization of the total entropy of matrix Y.

Subsequent transformations lead to the Kullback-Leibler criterion (Mansour et al 2000):

$$I(x_1,x_2,...,x_N)=\sum p(x_1,x_2,...,x_N)\cdot\log\{p(x_1,x_2,...,x_N)/p(x_1)p(x_2)...p(x_N)\} \quad (4.10)$$

The minimization of that function involves finding the root of its derivative, i.e. calculating the Kullback-Leibler divergence (Jung 2003; Mansour et al 2000, Torkkola 1996).

$$\frac{\partial I(y)}{\partial W}=-\frac{\partial}{\partial W}D[p(y_1,y_2,...,y_N)\,||\,p(y_1)p(y_2)...p(y_N)]= \quad (4.11)$$

$$=\frac{\partial}{\partial W}\int p(y_1,y_2,...,y_N)\cdot\log\frac{p(y\ ,y_2...,y_N)}{p(y_1)p(y_2)...p(y_N)}dy$$

Subsequent phases of ICA operations can be presented as follows (Karhunen 1996; Torkkola 1996, 1999):

- filtering out signal noise and removing the average value of the signal arriving from the microphones,
- separation into frames with the length determined by time (e.g. attack time) or by the spectral signal parameters (e.g. tracking the trajectory of fundamental frequency),
- the assumption about a usually random initial value of a demixing matrix coefficients,
- input data filtration using a demixing matrix in order to achieve source estimates,
- non-linear transformation of achieved estimate using a non-even, non-square and not too steep function (e.g. trigonometric functions tgh(*), log(cosh(*)), sigmoid),
- calculating ΔC using gradient methods or artificial neural networks,
- collecting another frame and returning to the second point of the algorithm,
- normalization of matrix C and its combination with matrix X in order to achieve the input data estimate.

Apart from neural networks other classifiers such as for example Bayesian or Hidden Markov Model can be used (Valpola et al 2003).

Independent Subspace Analysis (*ISA*)

The ISA (*Independent Subspace Analysis*) algorithm is a modification of the ICA algorithm with one important feature: the limitation that the number of sources must be smaller than the number of sensors has been removed. Another change in the ICA algorithm is the possibility of separating non-stationary signals. This becomes possible due to the use of dynamic sound components with tracking specific sources. The ISA ex-

pands the previous method with the identification of multi-component subspaces in sound sources using the STFT transform. It is assumed that sound signals are multi-dimensional when the number of microphones is greater than one; this, however, does not agree with the approach considered in the equation $x(k)=As(k)$, where the sources are considered one-dimensional.

The algorithm proposed by Casey uses the chart called an ixegram, which is a matrix of cross-entropies of independent components (Casey 2001). The ixegram is calculated by using the Kullback-Leibler divergence approximation. The Kullback-Leibler divergence is the measure of distance between two probability functions $P_a(u)$ and $P_b(u)$ for a random variable u. The ixegram is calculated according to the following relations (Casey 2001):

$$D(i,j) = \delta_{KL}(z_i,z_j) \tag{4.12}$$

$$\delta_{KL}(z_i,z_j) = KL(P_{zi}(u),P_{zj}(u)) \tag{4.13}$$

where: $i,j = 1\ldots n$, the KL divergence is determined according to the following formula:

$$KL(p(\hat{u}),q(\hat{u})) = \frac{1}{2}\int p(\hat{u})\cdot\log\left(\frac{p(\hat{u})}{q(\hat{u})}\right)\cdot d\hat{u} + \frac{1}{2}\int q(\hat{u})\cdot\log\left(\frac{q(\hat{u})}{p(\hat{u})}\right)\cdot d\hat{u} \tag{4.14}$$

If both probability functions are equal, the divergence will be 0. The ixegram matrix looks as follows (Casey 2001):

$$D = \begin{matrix} \delta_{KL}(z_1,z_1) & \delta_{KL}(z_1,z_2) & \ldots & \delta_{KL}(z_1,z_N) \\ \delta_{KL}(z_2,z_1) & \delta_{KL}(z_2,z_2) & \ldots & \delta_{KL}(z_2,z_N) \\ \ldots & \ldots & \ldots & \ldots \\ \delta_{KL}(z_N,z_1) & \delta_{KL}(z_N,z_2) & \ldots & \delta_{KL}(z_N,z_N) \end{matrix} \tag{4.15}$$

The final result of the algorithm operations, i.e. separated signals, comes from the separation of the ixegram.

Algorithms Based on Estimation and Resynthesis of Spectrum

The family of algorithms based on spectral analysis and synthesis is the largest and the most varied. There are algorithms using resynthesis of fundamental frequency and harmonic estimation (e.g. Serra 1997), as well as others, more complex ones, in which databases including the models of specific instruments are used. It has a great influence on the operations of

the algorithm in the case of the analysis of non-harmonic and transient sounds.

Sinusoidal Model

One may assume that each signal can be presented as a sum of sinusoids. It is obvious, however, that such assumption is useful only for stationary periodic signals.

$$\hat{s}(t) = \sum_{n=1}^{N} \sum_{k \in S_n} a_k \cos(2\pi f_k t + \Phi_k) \tag{4.16}$$

where:

N – the number of co-sounding sources

S_n – the set of harmonic components of the nth source

A_k, f_k, Φ_k – the amplitude, the frequency and the phase of subsequent harmonic components

The idea of sinusoidal modeling is connected with the use of sinusoids of frequencies and amplitudes variable in time to model harmonic signals. The extraction of sinusoids from the original signal involves the calculation of the STFT spectrum divided into blocks and a windowed signal, and then finding spectrum peaks. Then from the sonographic signal representation obtained in such a way, trajectory non-continuities are removed – ones which appeared as the result of vibrato and transients. In order to maintain trajectory, continuity interpolation is done in those points. In the next step the algorithm searches in each subsequent frame for a spectrum peak which is the most similar to the trajectory in a current frame. The result is the set of sinusoidal trajectories with frequency and amplitude variable in time. Usually the frequency of the components has a maximum limit, since the trajectories above 5 kHz are so small that it is difficult to detect them without mistakes.

In sinusoidal modeling, long windows with a large overlapping coefficient are used. It results from the fact that there are often two frequencies located close to each other in a signal. In such case the window length does not depend on the wavelength for a given frequency, but on a frequency difference. The non-stationary state of a signal limits the maximum length of a data block.

As mentioned before, human perception forces the use of consonants in music, which results in the overlaping of aliquots from many instruments. It is the greatest problem of sinusoidal modeling algorithms, which limits its usability. Dissonance intervals does not cause the overlapping of trajec-

tories at low frequencies, but it happens that their proximity causes estimation mistakes in the algorithm.

Many concepts of solving the problem have been invented. One of the most popular is defining the fundamental frequencies and then using them to find the frequencies of colliding components. With the information about the envelopes received from the sonogram, one can divide the energy of overlapping tones proportionally. In the final step of the algorithm the synthesis of separated signals by adding adjusted trajectories takes place. The method, although very intuitive, has its drawbacks, especially during the phase of separating a signal from a large number of sources. The number of overlapping and undetected harmonic components rapidly increases, generating significant mistakes.

Multipitch Estimation (MPE) Algorithm

The MPE (*Multipitch Estimation*) algorithm is similar to sinusoidal modeling and is, in a way, its expansion. It consists of two major repeated phases: estimation of a dominant harmonic component and the synthesis of a spectrum on the basis of the component, and subtracting it from a useful signal.

The first phase gives the best results after separating several sub-bands with spectrum maxima from the whole signal spectrum. In the next phase the results are synthesized and a global estimate is achieved. Such approach helps to avoid the mistakes resulting from a non-harmonic characteristic of the signal and offers benefits resulting from the increase of frequency resolution. Detailed scheme of the MPE algorithm was presented in the study of Klapuri et al. (2000).

Signal X_k is windowed using a Hamming window. In the same block signal processing takes place, which should remove unwanted noise from the signal and improve spectral quality of sound mixture. Improved spectrum $X_e(k)$ is the result of a logarithm used in a spectrum module followed by high pass filtration. In the algorithm proposed by Klapuri, Virtanen and Holm spectrum is divided into 18 logarithmically separated bands from 50 Hz to 6 kHz with triangular weighting windows applied (Klapuri et al 2000). Therefore it is the position similar to the one of the critical bands of human hearing. Sub-bands overlapping reaches 50%, resulting in the sum of windows equal 1.

In each of the sub-bands the probability vector $L_B(n)$ is calculated from the condition so that specific nth samples of the spectrum would maximize the probability. Samples $X_e(k)$ in band B are included in the range $k \in [k_B, k_B + K_B - 1]$, where k_B is the lowest sample of the spectrum, and K_B is their number in the sub-band (Klapuri et al 2000).

$$L_B = \max_{m \in M} \left\{ W(H) \sum_{h=0}^{H-1} X_e(k_B + m + hn) \right\} \tag{4.17}$$

where:

$m \in M$, $M = \{0, 1, ..., \nu-1\}$

$H = \{(K_B - m)/n]$ – the number of harmonic components of the sum

$W(H) = 0.75/H + 0.25$ – the normalization coefficient

In the final step the probabilities of the bands are consolidated and the result is the total $L(n)$; its maximum value is used to specify the fundamental functionality. At the output the following parameters are generated: the fundamental functionality F, the non-harmonic coefficient β and the complete spectrum of musical signal.

In order to completely remove a harmonic component from the signal, one needs to know its amplitude, frequency and phase. It is assumed that they are constant along the whole length of the frame. Therefore with the use of estimated parameters the spectrum is approximated in the proximity of the component, and then it is subtracted linearly from the spectrum of sound mixture. The spectrum-domain actions do not require so many operations as in the case of a time-domain, which still requires the transfer into a transform-domain and vice versa. The estimation of parameters, the calculation of the local spectrum module, and its subtraction are repeated for each component, in order to remove the appropriate elements from the mixture.

The problem of frequency overlapping takes place equally often as in the previously discussed sinusoidal model. Removing the whole coherent harmonic component from the spectrum becomes visible after several iterations, when remaining sounds are too distorted to produce any more useful iterations. One of the solutions used in this case is to perform spectrum smoothing before subtraction. The concept refers to psychoacoustic assumption that human ear links a series of components with one source easier when they have a smooth spectrum which decreases with the increase of frequency.

A considerable drawback of the MPE method is the sensitivity to vibrato effects, causing even very small changes of fundamental functionality. A strong advantage is the fact that the algorithm remains effective in a noisy environment or in the case of percussion sounds.

Spatial Filtration

The methods discussed in the following subsections are modifications of previously discussed blind separation algorithms. The difference is in the

use of beamforming algorithms. The objective of such algorithms is to locate sound sources in space and in the next step – to separate them. Therefore it makes the extraction from many sources even with a small number of microphones possible, which is extremely important in the case of commonly used stereophonic recordings.

Many algorithms have been developed to resolve spatial filtration problem such as for example classical delay and summation algorithms, superdirective arrays, adaptive algorithms and nonlinear frequency-domain beamformers. In some algorithms a narrow shape of the beam pattern, reduction of the noise power and improvement of the speech intelligibility in the presence of noise were achieved. The classic algorithms were based on adaptive signal filtering (Frost 1972; Griffiths and Jim 1982; Duvall 1983; Soede et al 1993). The basic purpose of an adaptive filter is to estimate the noise at any given moment of time and to subtract it from the useful signal. The noise estimation is done through the use of a correlated source of noise and a continuous modification of the filter parameters so that the output mean-square error can be minimized. The results shown by several authors were encouraging, but the obtained improvement of the signal-to-noise ratio (SNR) was not fully satisfactory for 2-microphone arrays.

An algorithm developed at the Multimedia Systems (former Sound and Vision Engineering) of Gdansk University of Technology proved that it is possible to obtain very narrow characteristics of a spatial filter which can be very useful for real-life applications (Czyzewski 2003; Czyzewski et al 1998, 2001; Kostek et al 1999; Lasecki et al 1999). The effectiveness of the developed algorithm was tested with the use of sounds representing the desired signal and some background noises: concurrent speech, white noise and harmonic tones. In all cases the desired signal was preserved at the algorithm output and signals from lateral direction between 15^0 and 90^0 were attenuated approx. 40 dB for frequencies below and equal to 1 kHz, whereas for frequencies larger than 1 kHz from 60 to 100 dB (Czyzewski 2003; Czyzewski et al 1998, 2001; Kostek et al 1999; Lasecki et al 1999).

In order to apply a spatial filtration (*beamforming*) signal samples should be collected in various points of the space. Spatial discrimination depends on the ratio of the size of the system aperture and the length of the signal wave. The bigger this ratio the better spatial discrimination. The objective of the space sampling is to achieve a beam pattern as exemplified in Fig. 4.2.

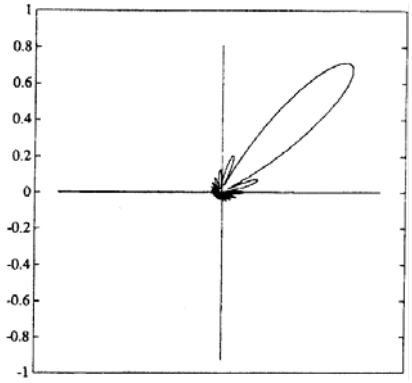


Fig. 4.2. Example of a beamforming signal pattern (vertical axis denotes gain) (Veen and Buckley 1988)

In recent years, many algorithms have been described in which the narrow shape of the beam pattern, reduction of the noise power and improvement of the speech intelligibility in the presence of noise were achieved. A classic beamformer denotes the linear combination of time samples as received by the particular transducers. The literature provides reference on two typical beamformer structures (see Fig. 4.3).

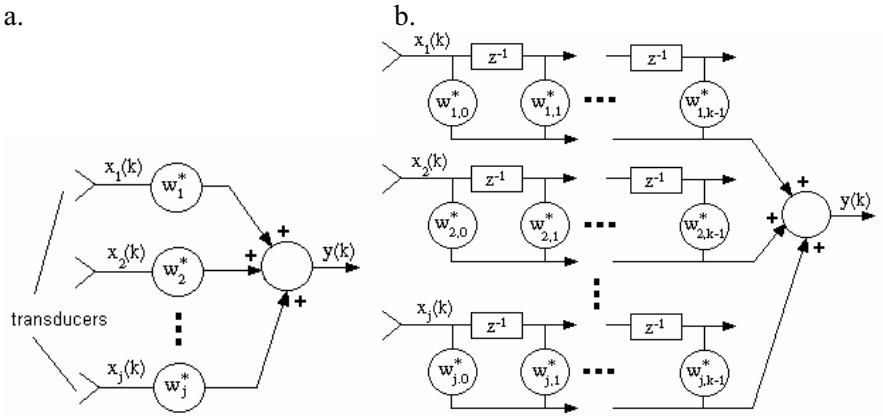


Fig. 4.3. Two typical beamformer structures, where: $x(k)$ – input signal, * - means the complex conjugate number, w_j – weights, z^{-1} – delays, $y(k)$ – output signal, j – is the number of channels, k – the number of delays in the particular channels (Veen and Buckley 1988)

The first structure (Fig. 4.3a) is used in the case of narrow band signals. In this case, the output signal is equal to the sum of the product of input signals and of weights that correspond to subsequent transducers. The sec-

ond structure is employed in the case of wide band signals. The output signal is obtained by the multiplication of delayed input signals and of the corresponding weights.

Fig. 4.4 illustrates how spatial sampling of a signal is done using an array of transducers. The figure shows the sampling process of a signal that is being propagated as a plane wave from the source located in direction (Θ). Given j transducers and k samples per a transducer in every moment of time, the propagating signal is sampled unequally in points $n=j \cdot k$. In addition, $t(\Theta)$ means time which elapses from the recording of the first sample in the first transducer until the recording of the last sample in the last transducer and is called the time aperture of observation for the given incidence angle (Θ). As the applied notation suggests, the time aperture is dependent on the angle (Θ).

Some other techniques were developed in order to perform spatial filtration in a more effective way. Examples of such algorithms will be shown in Chapter 5.4.

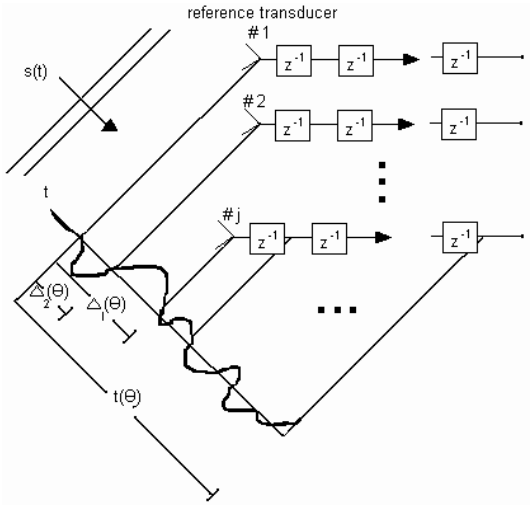


Fig. 4.4. Array with delay blocks realizing spatial and time sampling (Veen and Buckley 1988)

DUET Method

The DUET method is one of few techniques using spatial information in musical data separation. It is relatively simple, easily manipulated and, what is more important, gives satisfying results. Two stereophonic channels are used: left $x_L(t)$ and right $x_R(t)$ and the STFT transform is calculated

for both of them: $X_L(w,t)$ and $X_R(w,t)$. Two important functions are defined (Viste and Evangelista 2002):

Reference amplitude:

$$A(\omega,t)=\left|\frac{X_L(\omega,t)}{X_R(\omega,t)}\right| \tag{4.18}$$

and phase delay:

$$D(\omega,t)=\frac{\arg\left(\frac{X_L(\omega,t)}{X_R(\omega,t)}\right)}{\omega} \tag{4.19}$$

Placing the functions on two axes creates a two-dimensional histogram presenting spatial positioning of sources. It enables a very precise definition of the acoustic scene. Important advantages result from this fact. First, the sources do not need to have a harmonic character and can generate non-stationary or transient sounds. Second, there may be more sources than sensors, which was impossible in a family of blind separation algorithms.

However, a system without disadvantages does not exist. DUET is based on the assumption, that all sources are mutually orthogonal. In great majority of musical compositions the instruments every now and then play the same sounds or their octave transpositions, which is against the rule of orthogonal position of sources. After the appropriate separation of sources and after assigning specific frequencies to each of them, it may turn out, that some harmonic components disappear from some sources, and appear stronger in some others. The result might be an unpleasant sound, devaluating the effect of the algorithm operation.

There is a simple way of solving that problem, however it works only in the case of analysis performed on a stereophonic signal and when only two harmonic components overlap.

Using the formula for a mixed signal:

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}=\begin{pmatrix} h_0(t) & h_1(t) \\ h_2(t) & h_3(t) \end{pmatrix}*\begin{pmatrix} s_1(t) \\ s_2(t) \end{pmatrix} \tag{4.20}$$

and assuming that h_0 and h_3 are identical filters $\delta(t)$ and calculating the transform STFT the following result is achieved:

$$\begin{pmatrix} X_1(\omega,t) \\ X_2(\omega,t) \end{pmatrix}=\begin{pmatrix} 1 & H_1(\omega) \\ H_2(\omega) & 1 \end{pmatrix}\begin{pmatrix} S_1(\omega,t) \\ S_2(\omega,t) \end{pmatrix} \tag{4.21}$$

where:
x_i, X_i – a signal from the ith microphone and its spectrum
s_i, S_i – a signal from the ith source and its spectrum
h_{ij} – coefficients of the mixing matrix
$H_{ij}(w)$ – mixing filter coefficients

It is assumed that coefficients $H(\omega)$ have a constant value in the whole band and are integer complex numbers. In the next step the coefficients are estimated and then estimated spectral sources are calculated.

$$\begin{pmatrix} \hat{S}_1(\omega,t) \\ \hat{S}_2(\omega,t) \end{pmatrix} = \begin{pmatrix} 1 & -\hat{H}_1 \\ -\hat{H}_2 & 1 \end{pmatrix} \begin{pmatrix} X_1(\omega,t) \\ X_2(\omega,t) \end{pmatrix} \tag{4.22}$$

$$\begin{pmatrix} \hat{S}_1(\omega,t) \\ \hat{S}_2(\omega,t) \end{pmatrix} = \begin{pmatrix} 1-\hat{H}_1 H_2 & H_1-\hat{H}_1 \\ H_2-\hat{H}_2 & 1-\hat{H}_2 H_1 \end{pmatrix} \begin{pmatrix} S_1(\omega,t) \\ S_2(\omega,t) \end{pmatrix} \tag{4.23}$$

The closer the estimates are to real values, the smaller the presence of unwanted signal in the source. When they achieve optimum values, the components are proportionally divided between both sources. The situation is different for a monophonic version of the acoustic scene. In such a case only one sensor and several sources are available. Similar situation takes place when overlapping frequencies come from more than two sources. So far, several concepts of dealing with the above-mentioned problem have been researched, among them: removing, flattening, synthesis and separation of these components.

ESPRIT Algorithm

The ESPRIT (*Estimation of Signal Parameter via Rotational Invariance Technique*) algorithm is an efficient tool for estimating the direction, from which the signal comes. ESPRIT belongs to so-called DOA algorithms (*direction-of-arrival*). Generally some versions of this algorithm are used: standard ESPRIT, and Unitary-ESPRIT, in which the transfer from complex numbers to real numbers takes place significantly decreasing the number of calculations (Satayarak et al 2002). By exploiting invariances designed into the sensor array, parameter estimates are obtained directly, without the knowledge of the array response and without the computation or search of some spectral measures. The original formulation of ESPRIT assumes only one invariance in the array associated with each dimension of the parameter space. However, in many applications, arrays that possess multiple invariances (eg. uniform linear arrays, uniformly sampled time se-

ries) are employed (Swindlehurst et al 1992). Therefore another version of this algorithm is a so-called MI-ESPRIT (Swindlehurst et al 1992).

In the ESPRIT method L elements of the matrix are separated into two identical overlapping sub-matrices, each consisting of L-1 pairs of sensors dislocated by a known constant dislocation vector Δ, which expresses the reference distance. All angles in the algorithm are calculated in relation to this vector. The length of the vector is expressed using the length of wave Δ_0.

Several new techniques for DOA estimation using arrays composed of multiple translated and uncalibrated subarrays appeared. They do perform better than a single invariance implementation of ESPRIT, and are thus better suited for finding the initial conditions required by the MI-ESPRIT search. The new algorithms can be thought of as generalizations of the MUSIC, Root-MUSIC, and MODE techniques originally developed for fully calibrated arrays (Kusuma 2000; Swindlehurst et al 2001).

FED Algorithm

In literature one can find many algorithms in the context of music separation. It would be valuable to read papers related to the Music Information Retrieval domain. A part of this research was also reported by the author in Proceedings of IEEE (Kostek 2004b). Apart from the above reviewed methods, another algorithm will be presented here, namely a so-called Frequency Envelope Distribution (FED), developed at the Multimedia Systems Department (Kostek et al 2002a, 2002b). Automatic musical signal separation is the subject of the Ph.D. thesis of Dziubinski, a student supervised by the author (Dziubinski 2005). Since his work is well in progress, thus lately some new developments in the proposed algorithms have been engineered (Kostek et al 2004, 2005; Dziubinski et al 2005). However, only the basic algorithm will be presented here, which, as mentioned before, was also presented by the author in the Proceedings of IEEE.

The FED algorithm decomposes the signal into linear expansion of waveforms, called EMO – Envelope Modulated Oscillations that are a combination of complex exponential signals modulated by complex amplitude envelopes. These waveforms are sinusoidal signals with a time varying phase and amplitude and are chosen to best match the harmonic parts of the signal (Kostek et al 2002a). Such a representation is similar to the one introduced by Serra (1997), however here inner products representing the decomposition frequency are directly related to the decomposition frequencies, whereas Serra's approach is based on retrieving partials of the signal from the spectrogram matrix. The presented solution works faster, since inner products are calculated only for chosen frequencies, and the re-

trieving phase is based on windowed inner product. In addition, non-harmonic structures can also be represented by such waveforms. This is possible when amplitude and phase changes have a frequency higher than the sinusoidal signal frequency. In practice, it means that in such analysis the attention is set to consecutive spectral lines assuming that each line changes its amplitude and phase versus discrete time. Since the aim is to reconstruct intelligible audio data after the separation process in order to perform listening tests, that is why phase is retained in calculations. Oppenheim and Lim (1981) and McAulay and Quatieri (1990) pointed out in their studies the importance of phase in signal reconstruction.

The input signal can be represented as a sum of EMO structures (represented by the amplitude envelope and phase) and a residual signal (Kostek 2004b).

$$S[n] = \sum_{i=1}^{K} A_i[n] \cos\left(\frac{2n\pi f_i}{f_s} + \Theta_i[n]\right) + R_s[n] \tag{4.24}$$

where $S[n]$ is the input signal, K is the number of decomposition iterations, A_i refers to the amplitude envelope for the ith iteration, Θ_i denotes phase envelope for the ith iteration, and R_s is the residual signal.

The first step of the FED algorithm is the Power Spectrum Density (PSD) estimation of the input signal using the Welch's averaged, modified periodogram method (Deller et al 1993; Proakis and Manolakis 1999). The frequency of the maximum value of the PSD ($f_{\max}$) is treated as the frequency of the most energy carrying EMO structure. Next is the calculation of nodes that represent the amplitude envelope of the real and imaginary part of a complex exponential related to $f_{\max}$. Such an operation can be viewed as calculating inner products of the signal and the complex exponential divided into frames, where the inner product of each frame represents the amplitude value. First, signals are multiplied sample by sample:

$$S_m[n] = S[n] e^{j\frac{2n\pi f_{\max}}{f_s}} \tag{4.25}$$

where $S[n]$ is the input signal, and $S_m[n]$ refers to the signal multiplied sample by sample by a complex exponential of frequency $f_{\max}$.

Signal S_m is divided into frames of the same length as that of the complex exponential period and for each block frame the value is calculated:

$$K_i = \sum_{m=1}^{w(f_{\max})} S_m^i[n] \tag{4.26}$$

where K_i is the amplitude value for the *i*th block, $w(f_{\max})$ refers to frame length related to $f_{\max}$, and S_m^i is the *i*th frame of S_m signal.

The node value for the *i*th frame is an inner product of the input signal in the *i*th frame and the complex exponential in the *i*th frame. To obtain amplitude signals of the same size as this of the input signal, appropriate interpolation has to be performed. Cubic spline approximation provides interpolating curves that do not exhibit large oscillations associated with high degree interpolating polynomials (Rabiner et al 1976) and, thanks to its low computational complexity, seems to be the perfect tool for the task of amplitude envelope interpolation. In the next algorithmic step cubic spline interpolation is performed. It is also used to overcome the problem with phase unwrapping.

The first decomposition step is then performed:

$$R_s[n] = S[n] - A_1[n]\cos(\frac{2n\pi f_1}{f_s} + \Theta_1[n]) \tag{4.27}$$

where R_s is the residual signal, f_1 refers to frequency f_{max} for the first iteration.

Each iteration is computed identically assuming that a residual signal of the previous iteration becomes the input signal for the next iteration. However, if the same $f_{\max}$ is detected, a significantly shorter amplitude calculation frame has to be applied and the iteration is then repeated, assuming that most of the energy carrying frequencies phase is disturbed and does not preserve harmonic properties. In this case the EMO structure represents the non-harmonic part of the signal. Decomposition frequencies are chosen a priori for the FED. The first decomposition frequency is the fundamental frequency of the lower pitched instrument. Therefore, it is necessary to first employ a pitch estimation algorithm.

Since multipitch detection is not needed at this stage and one is interested in the lower instrument pitch only, an algorithm based on the correlation analysis seems to be well suited to carry out this task (Rabiner et al 1976; Kostek et al 2002b). However several modifications were applied to improve the accuracy of the algorithm according to the research done by Masuda-Katsuse (2001).

The average pitch of a chosen segment results in the first decomposition frequency. It is assumed that this frequency is the fundamental frequency of the lower pitched instrument. Frequencies of the first ten harmonics are then calculated and FED iterations are performed for those frequencies. Since FED iterations can represent harmonic or inharmonic parts of the signal, a modification of the FED was necessary in order to decompose

only harmonic parts. Such modification is achieved by allowing only relatively large windows for calculating envelopes for each EMO.

The first *K* harmonics of the lower pitched instrument, within each segment, can be represented as a sum of EMO structures and can be written in a simplified way as:

$$I_1[n] = \sum_{i=1}^{K} (\mathrm{Re}(\ EMO\ (S_m)^{fi}{}_i[n]) + \mathrm{Im}(\ EMO\ (S_m)^{fi}{}_i[n])) + \\ + R_{I_1}(S_m)[n] \tag{4.28}$$

where S_m is the *m*th segment of the input signal, I_1 is the extracted signal containing harmonic components of the lower pitched instrument, K is the number of iterations or the number of harmonics to be decomposed, *fi* is the frequency corresponding to the *i*th harmonic, $EMO(S_m)^{fi}$ refers to the *i*th Envelope Modulated Oscillation corresponding to the *i*th harmonic frequency, and $R_{I_1}(S_m)$ is the residual signal containing inharmonic components of both instruments and harmonics of the higher pitched instrument.

The pitch detection procedure is repeated for $R_{I_1}(S_m)$ resulting in Pitch Contour Signal. Further segmentation of S_m is carried out if necessary. FED decomposition is repeated for each segment of S_m. The first *K* harmonics of the higher pitched instrument can be represented as a sum of EMO structures:

$$I_2[n] = \sum_{i=1}^{K} (\mathrm{Re}(EMO(S_{m_p})^{fi}{}_i[n]) + \mathrm{Im}(EMO(S_{m\,p})^{fi}{}_i[n])) + \\ + R_{I_2}(S_{m_p})[n] \tag{4.29}$$

where S_{m_p} is the *p*th segment of the S_m, I_2 is the extracted signal containing harmonic components of the higher pitched instrument, *K* refers to the number of iterations or the number of harmonics to be decomposed, *fi* is the frequency corresponding to the *i*th harmonic, $EMO(S_{m_p})^{fi}$ denotes the *i*th Envelope Modulated Oscillation corresponding to the *i*th harmonic frequency, and $R_{I_2}(S_{m_p})$ is the residual signal containing inharmonic components of both instruments and harmonics of the lower pitched instrument.

Signal Decomposition

The segmentation of a sound based on Pitch Contour Signal (PCS) enables small fluctuations of pitch. Pitch for each segment is actually an average pitch within such a segment. This generalization does not produce large errors in the algorithm, since each EMO structure thanks to the envelope frequency modulation properties adapts itself to such small fluctuations.

In Fig. 4.5 an example of the Pitch Contour Signal is shown for instruments A and B mixed together. One segment of the input signal with constant pitch becomes the input signal for the FED decomposition algorithm. FED removes harmonics related to the detected pitch. The residual signal consists of harmonics from the higher pitched instrument. Based on the residual signal, pitch contour of the remaining instrument can be calculated. Since pitch of the higher instrument was not constant, further segmentation in this case would be required.

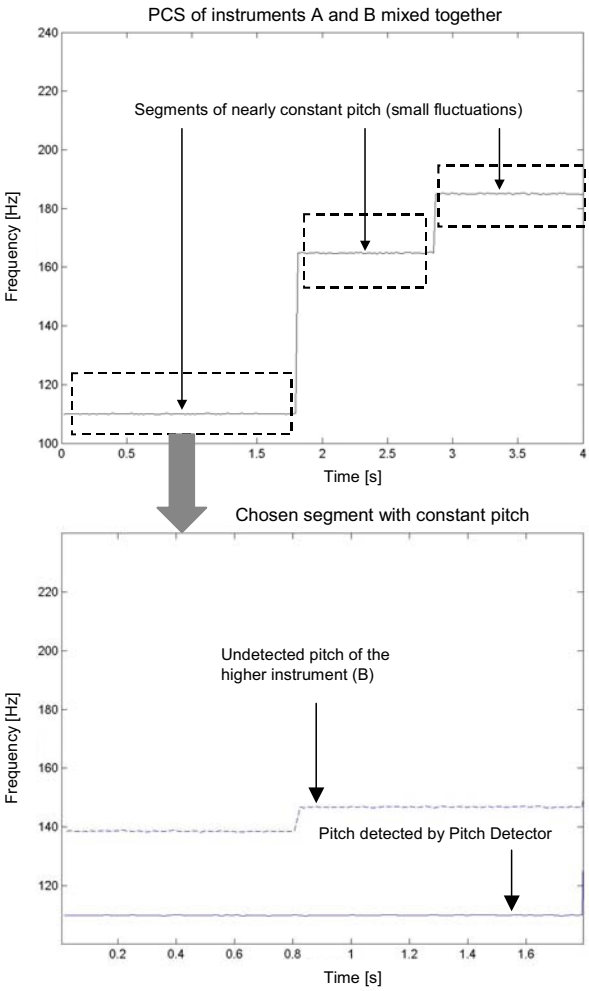


Fig. 4.5. Example of Pitch Contour Signal for instruments A and B mixed together

Harmonic Detection

Since fundamental frequencies of both instruments can be in harmonic relation, some of the harmonics from both instruments might share the same frequency. Frequencies of the coinciding harmonics can be easily calculated and eliminated for the task of sound recognition if pitch of both instruments is known, and eliminated for sound recognition tasks. Additionally, FED decomposition can be carried out for $R_{I_2}(S_{m_p})$ and

for $R_{I_1}(S_{m_p})$, since both residual signals do not contain coinciding frequencies.

The FED of the residual signals can be expressed in a simplified way as:

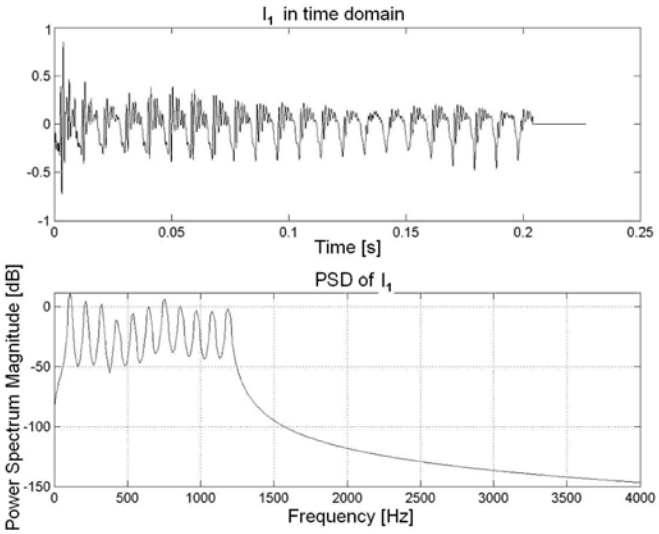
$$I'_2[n] = \sum_{i=1}^{K} (\mathrm{Re}(EMO(R_{I_1}(S_{mp})^{fi}{}_i[n]) + \mathrm{Im}(EMO(R_{I_1}(S_{mp})^{fi}{}_i[n])) \quad (4.30)$$

$$I'_1[n] = \sum_{i=1}^{K} (\mathrm{Re}(EMO(R_{I_2}(S_{mp})^{fi}{}_i[n]) + \mathrm{Im}(EMO(R_{I_2}(S_{m_p})^{fi}{}_i[n])) \quad (4.31)$$

where I'_2 is the higher pitched instrument signal in the *p*th segment, containing non-coinciding harmonics, and I'_1 is the lower pitched instrument signal in the *p*th segment, containing non-coinciding harmonics.

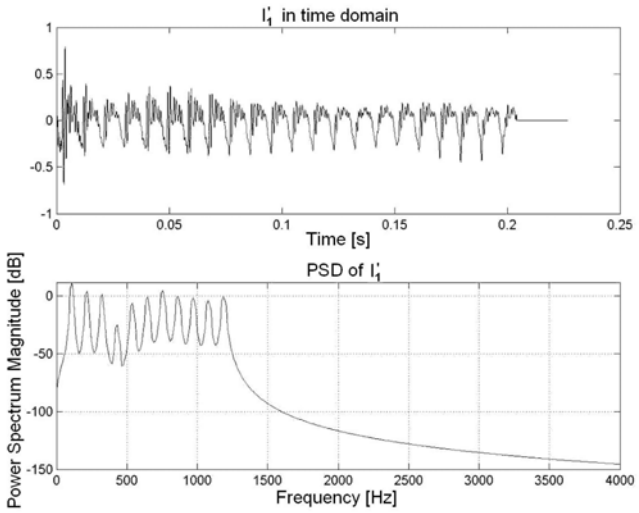
Fig. 4.6 shows I_1, I_2, I'_1 and I'_2 EMO representations resulting from one segment of a signal consisting of the mixed 448.8 Hz saxophone sound with 108.1 Hz cello sound. Fig. 4.6 contains both time- and frequency-domain plots of sounds after separation.

a. I_1

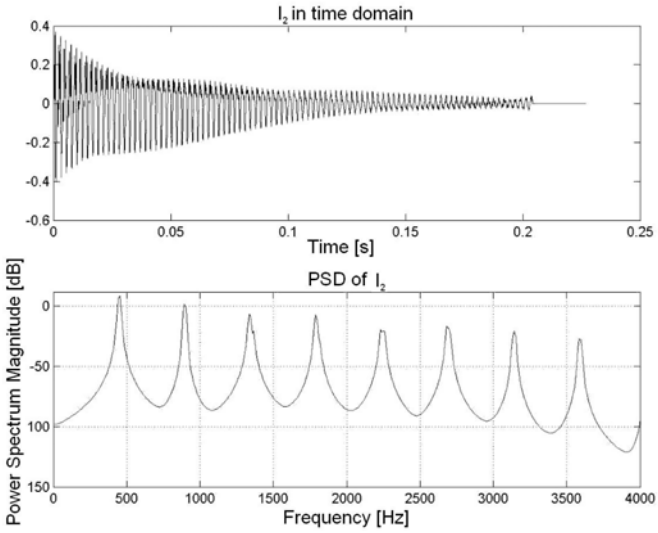


(Legend to Fig. 4.6, see next pages)

b. I'_1



c. I_2



(Legend to Fig. 4.6, see next page)

d. I'_2

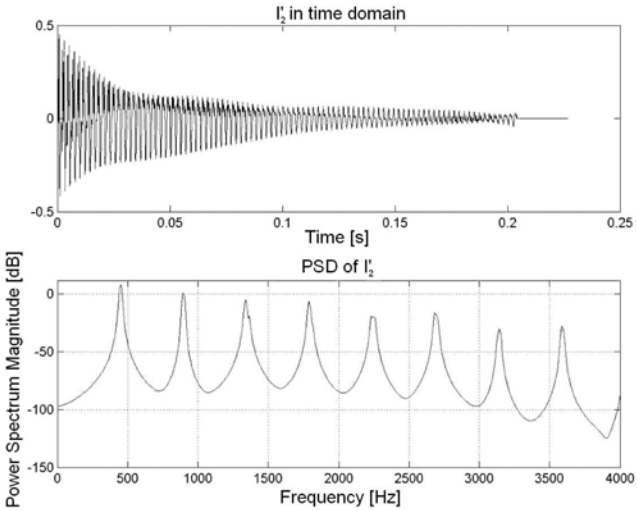


Fig. 4.6. Separated I_1, I_2, I'_1 and I'_2 signals (I_1, I'_1 refer to cello sound and correspondingly I_2, I'_2 to saxophone sound)

4.1.3 Automatic Separation of Musical Duets

For the purpose of checking the efficiency of the FED algorithm devoted to the task of musical duet separation, some musical instrument sound recognition experiments based on the ANN were used. The structure of this ANN was three-layered, consisting of correspondingly 8 (input layer), 20 (hidden layer) and 8 (output layer) neurons. A unipolar sigmoidal transfer function was used and a Back-Propagation training algorithm with momentum was applied during the training phase. The ANN was trained employing about 400 sound excerpts containing sounds of 8 instruments of differentiated musical articulation (Kostek et al 2002a). In addition, an optimization process of generalization properties was performed using another set of 400 sounds. It consisted in stopping the ANN training every time the mean square error appeared to increase.

Sounds that were separated employing the FED algorithm were then parametrized. I_1, I_2, I'_1 and I'_2 signal representations after separation were used for feature vector calculation and these FVs were then fed to the ANN. The sample answers of the neural network for pairs of musical

sounds: a clarinet and a trumpet and an English horn and a viola are presented in Tables 4.2-4.5. In Tables 4.2 and 4.3 the ANN answers are given for testing feature vectors derived from pairs of sounds before mixing (original sounds), and Tables 4.4 and 4.5 contain results for testing feature vectors resulted from parametrization of sounds after the separation based on the FED algorithm. The consecutive columns refer to instruments on which the ANN was trained. Rows correspond to sounds that were used in the ANN testing. Values contained in tables refer to the ANN output while testing particular sounds and they are highlighted in a bold font in the case of correct classification by the ANN. The following sounds were mixed in two groups in pairs as seen below:

trumpet A4 – clarinet A#4
trumpet B3 – clarinet A4
trumpet A3 – clarinet A#5
trumpet B3 – clarinet A#3
and
English horn A3 – viola A#3
English horn A#3 – viola A#4
English horn B4 – viola A3
English horn C#4 – viola A#4

Remark: B3 trumpet and viola A#4 sounds were used twice in the mix of sounds.

As seen in Tables 4.2 and 4.3 values at the output neurons corresponding to sounds being recognized were very close to 1, whereas the output of remaining neurons was close to 0. On the other hand, as seen in Tables 4.4 and 4.5 values at the output neurons corresponding to sounds being recognized were in some cases not so close to the value of 1, however, these neurons were the winning neurons in the output layer. In addition, some sounds were not recognized properly. The residual signal containing both inharmonic spectrum content and overlapping harmonics of the other sounds from the duet caused the erroneous answer, still making the recognition of one of the mixed sounds possible. For example, it can be observed that the recognition of the sounds of a viola and an English horn was much easier than the recognition of a trumpet and a clarinet sound for the ANN-based algorithm. In the first case the recognition system had some problems with octave related sounds, whereas in the second case only sounds of the first and fourth pairs were recognized properly.

Table 4.2. ANN output neuron answers for trumpet and clarinet sounds (original sounds)

Musical Instr./ ANN output	CLA A#3	CLA A4	CLA A#4	CLA A#5	TRU A3	TRU B3	TRU A4
VIOLA	0.0005	0	0.0003	0.0127	0	0	0.0034
ENG. HORN	0.032	0	0	0.3089	0.0047	0.0045	0.0039
FR. HORN	0.0053	0.0347	0.0453	0	0.0001	0.0001	0.0001
SAX	0.0176	0.2743	0.1415	0	0.0146	0.0343	0.0141
CLARINET	0.9888	0.9484	0.9178	0.7783	0.0006	0.0018	0.0001
BASSOON	0	0.0001	0.0001	0.0002	0	0	0
TROMBONE	0	0.0462	0.0176	0	0.0129	0.0011	0
TRUMPET	0	0	0	0	0.99	0.9877	0.9987

Table 4.3. ANN output neuron answers for viola and English horn sounds (original sounds)

Musical Instr./ ANN output	ENH A3	ENH A#3	ENH C#4	ENH B4	VIOLA A3	VIOLA A#3	VIOLA A4
VIOLA	0.6678	0.2621	0.4945	0.3158	0.9712	0.872	0.9916
ENG. HORN	0.9101	0.8726	0.808	0.892	0.0105	0.5759	0.0082
FR. HORN	0.001	0.0006	0.0007	0	0.0003	0.0003	0.0002
SAX	0	0	0	0	0	0	0
CLARINET	0.0024	0.0082	0.0003	0.3842	0.0102	0.0042	0
BASSOON	0	0	0	0	0	0	0
TROMBONE	0	0	0	0	0	0	0
TRUMPET	0	0	0	0	0	0	0

Table 4.4. ANN output neuron answers for trumpet and clarinet sounds (sounds after FED separation)

Musical Instr./ ANN output	TRU A4	TRU B3	TRU A3	TRU B3	CLA A#4	CLA A4	CLA A#5	CLA A#3
VIOLA	0.647	0.657	0	0	0.013	0.888	0.446	0.002
ENG. HORN	0.006	0.002	0.013	0	0	0.044	0.06	0.0001
FR. HORN	0.0001	0.0001	0.0009	0.491	0.099	0	0.0003	0.083
SAX	0.019	0.615	0.006	0.179	0.02	0.02	0	0.251
CLARINET	0	0	0.009	0.0001	0.866	0	0.008	0.942
BASSOON	0	0	0	0.0002	0	0	0	0
TROMBONE	0	0	0.0002	0.484	0.005	0	0	0
TRUMPET	0.956	0.01	0.931	0.762	0.138	0.224	0.313	0.085

Table 4.5. ANN output neuron answers for viola and English horn sounds (sounds after FED separation)

Musical Instr./ ANN output	ENH A3	ENH A#3	ENH B4	ENH C#4	VIOLA A#3	VIOLA A#4	VIOLA A3	VIOLA A#4
VIOLA	0.003	0	0.053	0.012	0.778	0.776	0.999	0.724
ENG. HORN	0.597	0.0002	0.848	0.97	0.0008	0.051	0.008	0.602
FR. HORN	0.001	0.115	0.0005	0.0001	0	0.0007	0.0002	0.004
SAX	0.0001	0.002	0.0001	0	0	0	0	0
CLARINET	0.011	0.021	0.003	0.004	0	0.184	0.0009	0.0006
BASSOON	0	0	0	0	0	0	0	0
TROMBONE	0	0.0008	0	0	0	0	0	0
TRUMPET	0.003	0	0.0004	0.0001	0	0	0	0

Comparison of Separation Algorithm Effectiveness Using Perceptron-based and SOM algorithms

More thorough analysis was conducted in the context of a separation algorithm quality by means of the perceptron-based and Self Organizing Map (SOM) algorithms (Cendrowski 2005). The effectiveness of three algorithms (FEDs, FEDr, *fixed-point* ICA) was tested on ten classes of musical instruments. Basically, the difference in these algorithms lays in the principle of the separation method. The difference in the first two mentioned algorithms consists in a synthesizing procedure, namely FEDs means that each synthesized sound was subtracted from the mix, thus creating a new input signal for the next separation stage (separation of the consecutive instrument). Separation order was established based on the average pitch of each instrument, starting from the sound with the lowest fundamental frequency. The last separated sound had the highest pitch. The synthesized signal contained only harmonic content. It should be noted that in the described experiments only two sounds were mixed. The second procedure (FEDr) consists in synthesizing a signal based on the residuum (created by subtracting all other sounds). In this procedure the residual signal, used for recognition, contained inharmonic (noisy) components, remaining after the subtraction of other sounds. The last mentioned algorithm is based on the Independent Component Analysis presented before in this Chapter.

The musical database (Musical Instrument Sounds) from the University of Iowa was used in the experiments described below. All instruments are posted on AIFF files on the website (http://theremin.music.uiowa.edu/MIS.html) of the University of Iowa. All samples are gathered in this database in mono, 16 bit, 44.1 kHz, the exception, however, is the piano, which is in stereo. Each note is approximately

2 seconds long and is immediately preceded and followed by ambient silence. Some instruments are recorded with and without vibrato. String instrument recordings include arco (bowed) and pizzicato (plucked). The only non-anechoic instrument is the piano, which was recorded in a small studio. Sound files consist of chromatic scales at three non-normalized dynamic levels, *pp*, *mf*, *ff*, however, in the experiments described samples with *ff* level were used.

From the whole set of sound files 296 sounds were extracted and transformed to the wav format. For separation sounds from the octave of C4 to B4 were chosen, resulting in 116 sound files for the 10 instruments. This means that twelve notes separated from one another by a semitone were used for all 10 instruments except double bass (sounds only up to G4 are available). The following 10 instrument classes were chosen for the experiments. They were as follows: saxophone (SAX), double bass (DB), trombone (TRO), bassoon (BAS), clarinet (CLA), piano (PI), trumpet (TRU), violin (VI), flute (FL), oboe (OBO).

All sounds were mixed with a C4 sound. This means, that for example for the modified FED algorithm 10 instruments multiplied by 116 minus 10 sounds (the mixture of C4 and C4 sounds were eliminated) resulted in 1150 sound samples. Only four sounds remained after the process of separation. On the whole, 4420 sound samples were available for testing the effectiveness of FED algorithms. On the other hand, for the ICA algorithm, only two sounds remain after the process of separation, which gives 2210 sounds for testing.

Table 4.6. Sound sample files used in experiments

Sound sample file name	
AltoSax.NoVib.ff.A#4.wav	flute.novib.ff.B4.wav
AltoSax.NoVib.ff.A4.wav	flute.novib.ff.C#4.wav
AltoSax.NoVib.ff.B4.wav	flute.novib.ff.C4.wav
AltoSax.NoVib.ff.C#4.wav	flute.novib.ff.D#4.wav
AltoSax.NoVib.ff.C4.wav	flute.novib.ff.D4.wav
AltoSax.NoVib.ff.D#4.wav	flute.novib.ff.E4.wav
AltoSax.NoVib.ff.D4.wav	flute.novib.ff.F#4.wav
AltoSax.NoVib.ff.E4.wav	flute.novib.ff.F4.wav
AltoSax.NoVib.ff.F#4.wav	flute.novib.ff.G#4.wav
AltoSax.NoVib.ff.F4.wav	flute.novib.ff.G4.wav
AltoSax.NoVib.ff.G#4.wav	oboe.ff.A#4.wav
AltoSax.NoVib.ff.G4.wav	oboe.ff.A4.wav
Bass.arco.ff.sulG.C#4.wav	oboe.ff.B4.wav
Bass.arco.ff.sulG.C4.wav	oboe.ff.C#4.wav
Bass.arco.ff.sulG.D#4.wav	oboe.ff.C4.wav
Bass.arco.ff.sulG.D4.wav	oboe.ff.D#4.wav
Bass.arco.ff.sulG.E4.wav	oboe.ff.D4.wav

Table 4.6. (Cont.)

Bass.arco.ff.sulG.F#4.wav	oboe.ff.E4.wav
Bass.arco.ff.sulG.F4.wav	oboe.ff.F#4.wav
Bass.arco.ff.sulG.G4.wav	oboe.ff.F4.wav
Bassoon.ff.A#4.wav	oboe.ff.G#4.wav
Bassoon.ff.A4.wav	oboe.ff.G4.wav
Bassoon.ff.B4.wav	Piano.ff.A#4.wav
Bassoon.ff.C#4.wav	Piano.ff.A4.wav
Bassoon.ff.C4.wav	Piano.ff.B4.wav
Bassoon.ff.D#4.wav	Piano.ff.C#4.wav
Bassoon.ff.D4.wav	Piano.ff.C4.wav
Bassoon.ff.E4.wav	Piano.ff.D#4.wav
Bassoon.ff.F#4.wav	Piano.ff.D4.wav
Bassoon.ff.F4.wav	Piano.ff.E4.wav
Bassoon.ff.G#4.wav	Piano.ff.F#4.wav
Bassoon.ff.G4.wav	Piano.ff.F4.wav
BassTrombone.ff.A#4.wav	Piano.ff.G#4.wav
BassTrombone.ff.A4.wav	Piano.ff.G4.wav
BassTrombone.ff.B4.wav	Trumpet.novib.ff.A#4.wav
BassTrombone.ff.C#4.wav	Trumpet.novib.ff.A4.wav
BassTrombone.ff.C4.wav	Trumpet.novib.ff.B4.wav
BassTrombone.ff.D#4.wav	Trumpet.novib.ff.C#4.wav
BassTrombone.ff.D4.wav	Trumpet.novib.ff.C4.wav
BassTrombone.ff.E4.wav	Trumpet.novib.ff.D#4.wav
BassTrombone.ff.F#4.wav	Trumpet.novib.ff.D4.wav
BassTrombone.ff.F4.wav	Trumpet.novib.ff.E4.wav
BassTrombone.ff.G#4.wav	Trumpet.novib.ff.F#4.wav
BassTrombone.ff.G4.wav	Trumpet.novib.ff.F4.wav
BbClar.ff.A#4.wav	Trumpet.novib.ff.G#4.wav
BbClar.ff.A4.wav	Trumpet.novib.ff.G4.wav
BbClar.ff.B4.wav	Violin.arco.ff.sulA.A#4.wav
BbClar.ff.C#4.wav	Violin.arco.ff.sulA.A4.wav
BbClar.ff.C4.wav	Violin.arco.ff.sulA.B4.wav
BbClar.ff.D#4.wav	Violin.arco.ff.sulD.D#4.wav
BbClar.ff.D4.wav	Violin.arco.ff.sulD.D4.wav
BbClar.ff.E4.wav	Violin.arco.ff.sulD.E4.wav
BbClar.ff.F#4.wav	Violin.arco.ff.sulD.F#4.wav
BbClar.ff.F4.wav	Violin.arco.ff.sulD.F4.wav
BbClar.ff.G#4.wav	Violin.arco.ff.sulD.G#4.wav
BbClar.ff.G4.wav	Violin.arco.ff.sulD.G4.wav
flute.novib.ff.A#4.wav	Violin.arco.ff.sulG.C#4.wav
flute.novib.ff.A4.wav	Violin.arco.ff.sulG.C4.wav

Quality testing of the separation algorithms includes the following steps:

- extraction of sound files from the database along with their fundamental frequency, mixing and separation, parameter extraction,
- neural network training on the basis of feature vectors extracted from sounds from octave,

- neural network testing on the basis of feature vectors extracted from sounds after separation,
- error analysis.

Feature vectors contain the following parameters, defined previously in Chapter 3.

Attack Time, *Log Attack Time* (*LAT*), *Log Attack Time* (duration from 0.1 to 0.5) ($LAT_{0.1\text{-}0.5}$), *Temporal Centroid* (*TC*), *Temporal Centroid* (*nTC*) normalized over time, content of even harmonics (h_{ev}), energy of harmonics 2 to 5 divided by the energy of the first harmonic (A_2:A_5), modified *Tristimulus* parameters (T_1:T_3), spectral centroid (*SC*), spectral centroid divided by the fundamental frequency (SC_f), mean value of the amplitudes of a harmonic computed in each frame (D_i), mean value of the amplitudes of a harmonic over time (H_i), standard deviation of the amplitudes of a harmonic over time (S_i), mean value of *Harmonic Spectral Centroid* (*mIHSC*), *Harmonic Spectral Centroid* denoted in [Hz], value of *Harmonic Spectral Spread* (*mIHSS*), value of *Harmonic Spectral Deviation* (*mIHSD*), *Audio Spectrum Envelope* (*ASE*), *normalized Audio Spectrum Envelope* over the whole energy of the sound (n*ASE*), value of *Audio Spectral Centroid* (*ASC*), value of *Audio Spectrum Spread* (*ASS*), *Spectral Flatness Measure* (*SFM*).

Feature vectors were tested in terms of instrument class separability, first. Various configurations of parameters in feature vectors were also tested by neural networks (perceptron-based). Typical parameters were used in training of neural networks. Levenberg-Marquardt optimization algorithm was used. During the quality testing phase the best results were obtained for the following content of the feature vector: [*TC*, *nTC*, *LAT*, A_2:A_5, TR_1:TR_3, h_{ev}, *SC*, *HSC*, *HSS*, H_1:H_{10}, S_1:S_{10}, *mIHSD*, *mASC*, *mASS*] (see results in Table 4.7). The number of properly classified instrument sounds can be found along the diagonal of the table. All results are obtained on the basis of feature vectors extracted from a sound sample after separation was performed.

Table 4.7. Quality of algorithms based on the classification of separated sounds performed by neural network (perceptron-based) [%]

FEDs	SAX	DB	TRO	BAS	CLA	PIA	TRU	VI	FLU	OBO
SAX (225)	**83.11**	2.22	0.89	0	0.44	0.44	0	8.00	4.00	0.89
DB (185)	36.76	**31.89**	15.14	0	0	2.16	0.54	8.11	1.62	3.78
TRO (225)	9.78	0.89	**68.00**	0	0	0	11.11	4.44	3.11	2.67

Table 4.7. (Cont.)

BAS (225)	25.78	8.00	4.00	**46.22**	0.89	0	0	4.89	9.78	0.44
CLA (225)	19.56	7.11	10.67	1.33	**52.00**	0.44	0	5.78	0	3.11
PIA (225)	8.44	3.11	1.78	2.67	0	**69.33**	0	9.78	0	4.89
TRU (225)	4.00	0	2.22	0	0	0	**45.78**	0.89	0	47.11
VI (225)	9.78	5.33	0	0	0	0	0	**83.11**	0.44	1.33
FLU (225)	62.67	0.44	0	0	0	0.89	0	4.44	**30.22**	1.33
OBO (225)	2.22	6.22	1.33	0	0	0	0.44	0.44	7.56	**81.78**
FEDr	SAX	DB	TRO	BAS	CLA	PIA	TRU	VI	FLU	OBO
SAX (225)	**72.44**	3.11	4.00	0.89	1.33	1.33	0	11.56	3.56	1.78
DB (185)	46.49	**15.14**	13.51	1.08	0	1.62	5.41	4.32	1.62	10.81
TRO (225)	20.44	2.67	**41.78**	0.44	0	0.89	5.33	6.67	0.44	21.33
BAS (225)	33.78	15.11	5.33	**30.67**	0.89	1.33	0	7.11	4.44	1.33
CLA (225)	44.44	9.78	6.67	0.44	**25.33**	0.44	0	4.44	0	8.44
PIA (225)	18.67	7.56	0.44	1.78	0	**45.78**	0	4.44	0	21.33
TRU (225)	21.33	1.78	5.78	0	0	0	**53.33**	6.67	0	11.11
VI (225)	20.44	2.22	1.33	0	0	0	0	**63.11**	2.22	10.67
FLU (225)	49.33	6.22	3.56	0.44	0.89	1.33	0	4.89	**22.67**	10.67
OBO (225)	18.22	11.11	1.33	0	0	0	0	6.22	0.44	**62.67**
ICA	SAX	DB	TRO	BAS	CLA	PIA	TRU	VI	FLU	OBO
SAX (225)	**70.00**	4.40	0	0.40	0.40	8.00	0	3.20	0	13.60
DB (185)	59.32	**11.02**	9.32	0.85	0	2.54	0	5.08	0	11.86
TRO (225)	6.12	1.22	**72.65**	0	0	5.31	8.57	2.04	1.22	2.86
BAS (225)	34.66	11.16	4.78	**25.10**	0	5.98	0.80	1.99	3.59	11.95
CLA (225)	24.90	5.22	1.61	0.80	**44.58**	5.62	1.20	2.01	0	14.06
PIA (225)	7.84	1.96	0.98	5.88	0	**65.69**	0.98	9.80	0	6.86

Table 4.7. (Cont.)

TRU (225)	3.73	0	0.41	0	0	6.22	**88.80**	0.41	0	0.41
VI (225)	16.80	0.39	0.78	0	0	5.86	0	**73.05**	0.39	2.73
FLU (225)	41.80	6.25	0.78	0.39	0	6.25	0.39	3.52	**14.06**	26.56
OBO (225)	5.79	7.44	5.37	0.41	0	4.96	0	2.48	1.65	**71.90**

It is interesting to analyze in which way the separation quality depends on a harmonic relationship in sounds that are mixed. In Fig. 4.7 such a dependence is shown. Such a dependence is especially noticeable for sounds of the same pitch, differences that occur are of order of dozen of cents. One can observe worsening of the results for the fifth and the fourth, where the ratio between frequencies is simple. The best results are obtained for the FEDs algorithm. However, it should be remembered that such an analysis was done on the basis of parameters, and, in addition, the choice of parameters was such as to diminish individual harmonic influence on the classification results, thus either normalization or energy ratios were computed.

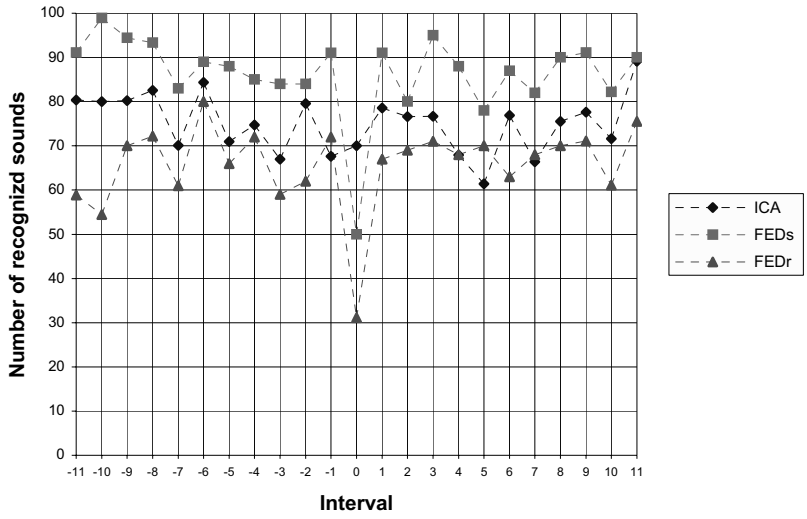


Fig. 4.7. Dependence of separation quality on harmonic relationship

Another interesting question arises as to which pair of instruments returns the best/worst results after separation is performed. The results of

such an analysis are shown in Table 4.8. Columns denote instruments that should be identified as such, rows show instruments that were added to the mixture, percentile denote correct classification. Mean and standard deviation values are also included.

Table 4.8. Influence of instrument used in mixture on quality of separation

FEDs	SAX	DB	TRO	BAS	CLA	PIA	TRU	VI	FLU	OBO	Mean
SAX	81.8	47.4	95.7	91.3	69.6	95.7	91.3	100	100	91.3	86.41
DB	94.7	35.7	94.7	89.5	73.7	89.5	89.5	100	100	94.7	86.2
TRO	87	73.7	95.5	87	82.6	95.7	100	100	100	100	92.15
BAS	95.7	26.3	87	95.5	78.3	95.7	91.3	95.7	100	95.7	86.12
CLA	47.8	47.4	95.7	87	77.3	100	95.7	95.7	100	95.7	84.23
PIA	87	57.9	95.7	100	73.9	100	100	100	100	100	91.45
TRU	56.5	36.8	82.6	78.3	78.3	87	100	100	100	91.3	81.08
VI	73.9	36.8	95.7	78.3	73.9	73.9	87	90.9	95.7	82.6	78.87
FLU	69.6	52.6	87	82.6	69.6	91.3	87	95.7	100	87	82.24
OBO	82.6	57.9	82.6	91.3	78.3	100	100	91.3	100	95.5	87.95
										Standard deviation	4.3
FEDr	SAX	DB	TRO	BAS	CLA	PIA	TRU	VI	FLU	OBO	Mean
SAX	59.1	84.2	100	82.6	82.6	73.9	87	91.3	73.9	91.3	82.59
DB	42.1	28.6	10.5	42.1	42.1	26.3	21.1	84.2	57.9	57.9	41.28
TRO	73.9	78.9	90.9	78.3	78.3	91.3	82.6	100	91.3	69.6	83.51
BAS	73.9	36.8	47.8	81.8	69.6	52.2	95.7	95.7	91.3	78.3	72.31
CLA	60.9	68.4	100	65.2	86.4	65.2	87	95.7	73.9	91.3	79.4
PIA	60.9	52.6	73.9	56.5	78.3	18.2	78.3	91.3	82.6	87	67.96
TRU	69.6	68.4	87	43.5	65.2	91.3	72.7	91.3	82.6	56.5	72.81
VI	26.1	26.3	47.8	39.1	56.5	21.7	43.5	81.8	78.3	43.5	46.46
FLU	56.5	73.7	26.1	69.6	73.9	8.7	87	87	68.2	82.6	63.33
OBO	78.3	52.6	91.3	91.3	65.2	87	87	100	100	90.9	84.36
										Standard deviation	15.2
ICA	SAX	DB	TRO	BAS	CLA	PIA	TRU	VI	FLU	OBO	Mean
SAX	50	90	92	36	52.2	66.7	91.7	93.1	52.6	100	72.43
DB	57.1	21.4	57.1	44	9.7	72.7	86.4	80	29.4	78.3	53.61
TRO	100	70	90.9	81.5	80.8	87.5	100	95.7	100	80.8	88.72
BAS	42.9	84.6	100	68.2	71.4	100	87.5	100	73.9	87.5	81.6
CLA	47.8	14.3	95	52	72.7	85.7	100	92.3	30.8	91.3	68.19
PIA	53.5	75	55.3	57.9	56.4	72.7	52.6	56.4	55	62.9	59.77
TRU	100	93.8	76.9	95.5	95.8	62.5	100	100	91.7	100	91.62
VI	94.1	92.3	95.7	100	95	100	100	90.9	91.7	100	95.97

Table 4.8. (Cont.)

FLU	18.5	25	95.8	52.2	40	66.7	100	100	36.4	87.5	62.21
OBO	76.9	93.3	95	81.8	87	81.8	100	100	86.4	68.2	87.04
									Standard deviation		14.9

Generally, the most stable algorithm was FEDs. The standard deviation was equal to 4.3. Also, in the case of this algorithm, the best results happened for a flute. The worst results were obtained for a double bass, independently of the algorithm used.

Below, results obtained for SOM algorithms are shown. SOM training parameters were as follows: neighborhood function – bubble; a batch training method; shape: hexagonal grids, *rough-fine tuning* method applied: training constant during *rough* phase: 0.5, training constant during *fine* tuning: 0.05; dimensions of maps are 10 by 10 units; initial training radius: 1.25, final training radius: 1. Results of quality testing of separation algorithms on the basis of the classification of sounds after separation done by the SOM algorithm are gathered in Tables 4.9 and 4.10. Columns represent clusters, to which an instrument was classified, consecutive rows show to which clusters instruments should be classified, '0' means that sounds were assigned to clusters represented by any instrument class.

Table 4.9. Algorithm effectiveness based on the classification of separated sounds performed by SOM

FEDs	'0'	SAX	DB	TRO	BAS	CLA	PIA	TRU	VI	FLU	OBO
SAX	5	185	2	0	5	16	0	0	12	0	0
DB	127	13	27	5	1	8	0	1	3	0	0
TRO	16	0	11	194	0	1	0	0	1	0	2
BAS	4	3	1	0	217	0	0	0	0	0	0
CLA	7	13	5	0	5	192	0	0	3	0	0
PIA	221	0	0	1	0	1	0	0	0	0	2
TRU	8	0	0	0	0	0	0	217	0	0	0
VI	5	0	0	0	0	0	0	0	220	0	0
FLU	12	0	1	11	0	0	0	0	0	201	0
OBO	4	0	0	0	0	1	0	0	0	0	220
FEDr	'0'	SAX	DB	TRO	BAS	CLA	PIA	TRU	VI	FLU	OBO
SAX	47	163	1	1	3	9	0	0	1	0	0
DB	45	11	93	16	0	10	2	0	2	6	0
TRO	38	0	0	177	2	0	0	8	0	0	0

Table 4.9. (Cont.)

BAS	94	2	0	0	127	0	0	0	2	0	0
CLA	42	16	1	1	7	157	0	0	0	1	0
PIA	145	9	5	32	6	0	27	0	1	0	0
TRU	19	0	0	0	0	0	0	206	0	0	0
VI	3	1	10	2	0	0	0	0	209	0	0
FLU	25	0	0	2	0	0	0	0	97	101	0
OBO	22	0	0	8	0	0	0	0	0	0	195
ICA	'0'	SAX	DB	TRO	BAS	CLA	PIA	TRU	VI	FLU	OBO
SAX	60	163	5	8	0	0	1	1	2	6	4
DB	21	67	17	7	0	0	0	0	0	6	0
TRO	15	12	11	201	0	0	4	0	1	1	0
BAS	39	23	1	19	141	1	1	9	5	8	4
CLA	45	10	1	19	0	154	1	1	3	9	6
PIA	68	3	1	4	0	0	24	1	0	0	1
TRU	16	4	0	3	0	0	1	215	0	0	2
VI	26	13	0	16	0	0	0	1	200	0	0
FLU	57	5	9	17	0	0	1	2	8	154	3
OBO	17	5	1	8	2	0	1	1	1	1	205

Table 4.10. Quality of algorithms based on the classification of separated sounds performed by SOM [%]

FEDs	'0'	SAX	DB	TRO	BAS	CLA	PIA	TRU	VI	FLU	OBO
SAX	2.2	82.2	0.9	0	2.2	7.1	0	0	5.3	0	0
DB	68.6	7	14.6	2.7	0.5	4.3	0	0.5	1.6	0	0
TRO	7.1	0	4.9	86.2	0	0.4	0	0	0.4	0	0.9
BAS	1.8	1.3	0.4	0	96.4	0	0	0	0	0	0
CLA	3.1	5.8	2.2	0	2.2	85.3	0	0	1.3	0	0
PIA	98.2	0	0	0.4	0	0.4	0	0	0	0	0.9
TRU	3.6	0	0	0	0	0	0	96.4	0	0	0
VI	2.2	0	0	0	0	0	0	0	97.8	0	0
FLU	5.3	0	0.4	4.9	0	0	0	0	0	89.3	0
OBO	1.8	0	0	0	0	0.4	0	0	0	0	97.8
FEDr	'0'	SAX	DB	TRO	BAS	CLA	PIA	TRU	VI	FLU	OBO
SAX	20.9	72.4	0.4	0.4	1.3	4	0	0	0.4	0	0
DB	24.3	5.9	50.3	8.6	0	5.4	1.1	0	1.1	3.2	0
TRO	16.9	0	0	78.7	0.9	0	0	3.6	0	0	0
BAS	41.8	0.9	0	0	56.4	0	0	0	0.9	0	0

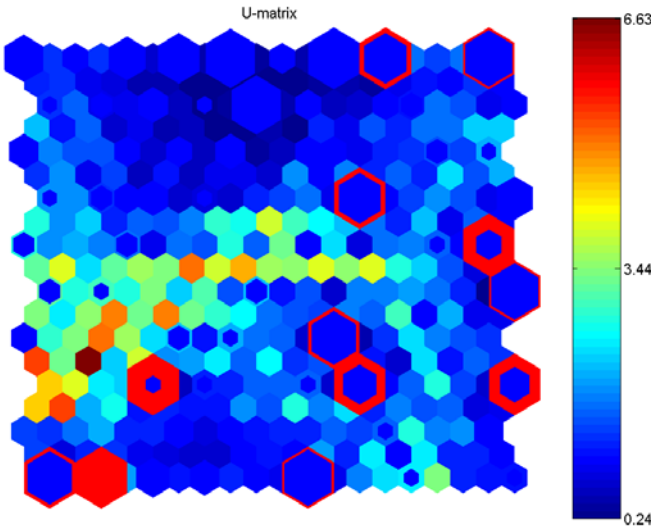
Table 4.10. (Cont.)

CLA	18.7	7.1	0.4	0.4	3.1	69.8	0	0	0	0.4	0
PIA	64.4	4	2.2	14.2	2.7	0	12	0	0.4	0	0
TRU	8.4	0	0	0	0	0	0	91.6	0	0	0
VI	1.3	0.4	4.4	0.9	0	0	0	0	92.9	0	0
FLU	11.1	0	0	0.9	0	0	0	0	43.1	44.9	0
OBO	9.8	0	0	3.6	0	0	0	0	0	0	86.7
ICA	'0'	SAX	DB	TRO	BAS	CLA	PIA	TRU	VI	FLU	OBO
SAX	24	65.2	2	3.2	0	0	0.4	0.4	0.8	2.4	1.6
DB	17.8	56.8	14.4	5.9	0	0	0	0	0	5.1	0
TRO	6.1	4.9	4.5	82	0	0	1.6	0	0.4	0.4	0
BAS	15.5	9.2	0.4	7.6	56.2	0.4	0.4	3.6	2	3.2	1.6
CLA	18.1	4	0.4	7.6	0	61.8	0.4	0.4	1.2	3.6	2.4
PIA	66.7	2.9	1	3.9	0	0	23.5	1	0	0	1
TRU	6.6	1.7	0	1.2	0	0	0.4	89.2	0	0	0.8
VI	10.2	5.1	0	6.3	0	0	0	0.4	78.1	0	0
FLU	22.3	2	3.5	6.6	0	0	0.4	0.8	3.1	60.2	1.2
OBO	7	2.1	0.4	3.3	0.8	0	0.4	0.4	0.4	0.4	84.7

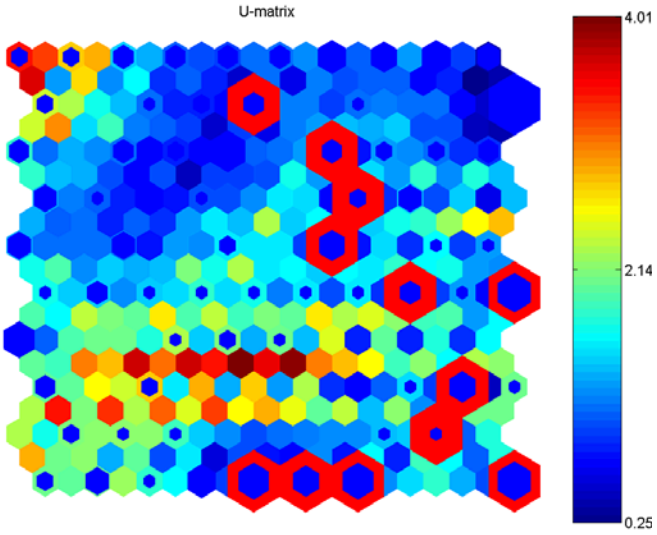
In addition, results obtained were visualized in a form of maps. The main aim of such an analysis was to show grouping of feature vectors extracted from sound samples before separation (reference vectors) and after separation, and in addition, to present topology of neurons forming clusters of instrument classes. The size and the shape of the SOM are of importance, because the distance between all classes should be minimal. In the case of experiments carried out the network topology was chosen as square. An example of such an analysis is shown in Fig. 4.8. It shows clustering density in different regions of data space for trumpet sound samples before and after separation. The size of neurons and their density reflect BMUs (*best matching units*) and the number of input vectors. The distance between neurons is determined by the unified distance matrix U of the size $2N$-1x$2M$-1, where N and M are sizes of the SOM. Elements of matrix U reflect distance between neurons. In a colored version of such SOMs red color illustrates reference vectors, and dark blue reflects feature vectors associated with sound samples after separation. Quality of separation is associated with the distance between neurons representing reference vectors and those extracted from sounds after separation, regardless of their placement in the organized map. As observed in Fig. 4.8 the best situation happened for the ICA algorithm, where all neurons belonging to reference vectors tend to immerse in the associated with feature vectors after separation.

Another kind of visualization of results obtained by the SOM is shown in Fig. 4.9. In this case all parameter values should be normalized with regard to their variances.

a. FEDs



b. FEDr



(Legend to Fig. 4.8, see next page)

c. ICA

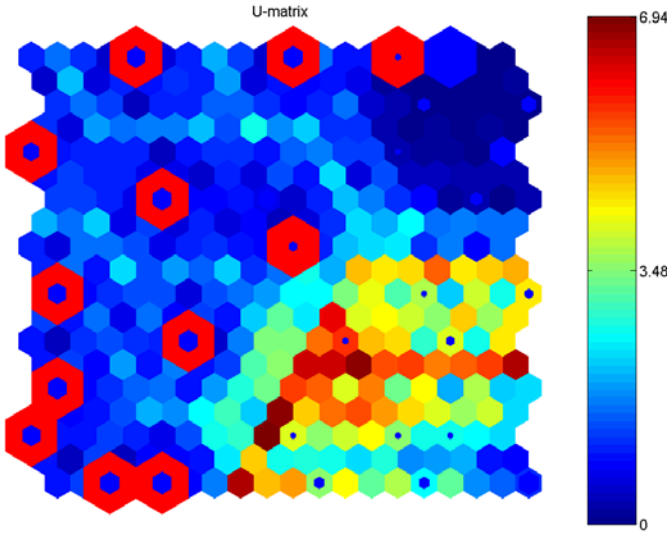
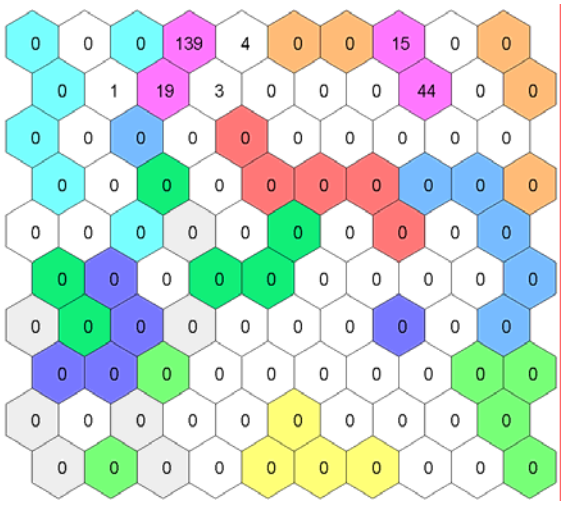


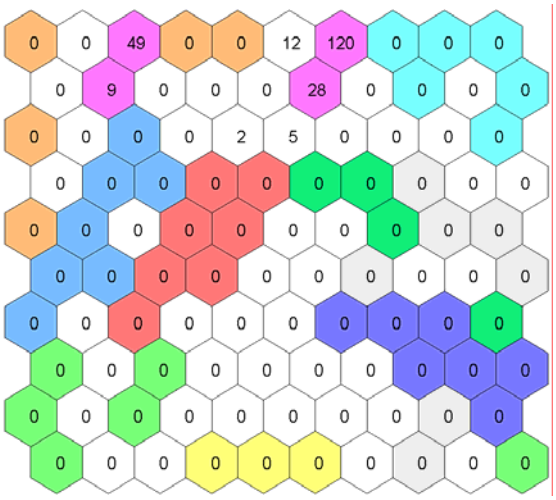
Fig. 4.8. Display of clusters for a trumpet before and after separation, the size and the intensity of gray color reflect classification of feature vectors to appropriate clusters

a. FEDs



(Legend to Fig. 4.9, see next page)

b. FEDr



c. ICA

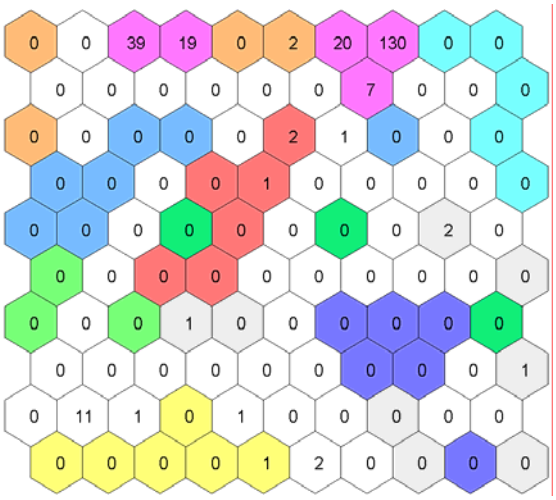


Fig. 4.9. Display of clusters of instruments, numbers show classification of feature vectors to the clusters (neurons associated with trumpet sound samples are placed in the upper part of figures, the first and second rows in the case of FED algorithms, in addition, a neuron in the third row for the ICA algorithm)

In Fig. 4.9 clusters reflecting the placement of reference vectors associated with trumpet sound samples obtained for all three algorithms are in the upper part (the first and second "rows" of neurons), these clusters are separated by empty space between them. Nearly all feature vectors extracted from sound samples after separation are gathered in these two areas

(see numbers shown in clusters associated with the trumpet). All three maps tend to coincide with this observation. However, a question remains whether, such an analysis reflects the actual structure of data, or whether it is an artefact of the parameter choice made by the researcher.

Lately, more thorough analysis of the automatic recognition of musical sounds after separation was carried out. Also, some new algorithms and procedures were introduced (Dziubinski et al 2005). The results are very satisfying for the engineered separation algorithms. They perform successful recognition of sounds after separation, even in cases when large amount of harmonic partials overlap, and sounds have overlapping transients in the mixture. Top algorithm-procedure combination allowed the ANN to correctly classify 97.4% of sound samples. This result is very encouraging, unfortunately with little room for improvement. In addition, for the purpose of recognition different feature vectors were created. All performed experiments proved that MPEG-7 descriptors alone are not sufficient for the classification of sounds separated with the given algorithm. Thus it was important to search for a set of descriptors specifically suitable for this purpose. In addition, it was shown that the manipulation of the attributes contained in feature vectors used for the ANN training and then for recognition has a significant influence on the results of recognition, and at the same time on the evaluation of separation techniques. It is important to mention that different descriptors might be necessary in more extensive tests, in which more than four instruments would be contained in a mix. In addition, real polyphonic performance (not artificially mixed isolated signals as in the examples presented) should be analyzed to provide more complete results.

4.2 Musical Phrase Analysis

In many existing Music Information Retrieval systems a musical phrase is a basis for formulating a query to such systems. Because of the increasing number of MIR systems, only some examples will be cited in this Chapter, therefore this review should not be treated as a systematic analysis of all existing Music Information Retrieval systems. A more systematic review on Music Information Retrieval systems is available from Typke's home page (http://mirsystems.info/).

The first introduced systems were based on extracting content-related features from symbolic musical data. Usually, a query was based on the sequence of the MIDI-transcribed notes introduced by a MIDI keyboard. Such a system created by Hawley (1990) enabled searching of the exact

sequence of notes. In practical applications such a condition cannot be easily fulfilled, thus in 1992 Wu and Manber engineered an algorithm better suited to real applications. Later, McNab et al. (1996a, 1996b) proposed a system (MELDEX) in which a query was introduced as an acoustical signal. Thus, the epoch of QBH (Query-by-Humming) systems appears. In most cases QBH systems utilize pitch contours to represent melodies. Apart from newer versions of such systems, they seldom used rhythm in their melody representation. Another group of systems concerns extracting content-related features from acoustical or audio signals. Two other groups may be referred to as extracting reference-related features from symbolic data, and reference-related features from symbolic musical data.

The MARSYAS (**M**usic**A**l **R**esearch **S**ystem for **A**nalysis and **S**ynthesis) or (**M**usical **A**nalysis and **R**etrieval **SY**stems for **A**udio **S**ignals) was designed and developed in Princeton University by Tzanetakis and Cook. One of the main characteristics of the system is that it supports many feature extraction schemes, among others, STFT-based features, Mel-Frequency cepstral coefficients, Linear-prediction coefficients, MPEG-7-based features, Discrete Wavelet Transform-based features. Other applications can be cited, such as: Themefinder application developed at Stanford University, the TuneServer project by Prechelt and Typke from the University of Karlsruhe (1997), the MiDiLiB project by the University of Bonn, etc. The already mentioned MELDEX (**MEL**ody in**DEX**) application created by McNab, Smith, Bainbridge and Witten from the University of New Zealand (1996b) was probably the first one introduced in the Internet. The system corpora consist of folk melodies (http://mirsystems.info/;http://213.133.111.178/Rntt/tuneserver_tochi2001.pdf; http://www.dlib.org/dlib/may97/meldex/05witten.html).

In 1995 Ghias, Logan, Chamberlin and Smith (1995) from Cornell University proposed a QBH (Query-by-Humming) system, in which the search is conducted using a melody introduced as an acoustical signal. The system architecture consisted of three main modules, namely: pitch-tracking module, melody database, query engine. After a melody is introduced to the microphone, it is registered, discretized and then pitch is detected. A melodic pitch contour creates the query. The system returns a list of melodies that are similar to the one searched for. All files are gathered in the MIDI format and they create a flat-type database. In a simple contour representation, a note in a piece of music is classified as a note-to-note movement in one of three ways: it is either a repetition of a previous note (R), or it is higher than a previous note (U), or it is lower than a previous note (D). Thus, the piece can be converted into a string with a three-letter alphabet (U, D, R). In the application mentioned, the Baesa-Yates and Perleberg's (1992) algorithm was adopted for the approximate matching with

k mismatches. In many QBH applications a 3-level contour representation was also used (in some systems, however, larger than 3). A variety of techniques are used in classification schemes, among others: dynamic programming, HMM (Hidden Markov Model), Gaussian Mixture Model, tree, etc. may be cited.

In 1997 Foote proposed the TreeQ system (Foote 1997), in which query results are based on the similarity measure. An MMI (*Maximum Mutual Information*) tree is used in this system. It is capable of being pruned to ignore irrelevant information. In Fig. 4.10 the basic structure of the system is shown, and, in addition, in Fig. 4.11 the principles of testing the database is shown. The QTree algorithm proposed by Foote was later adapted by other researchers.

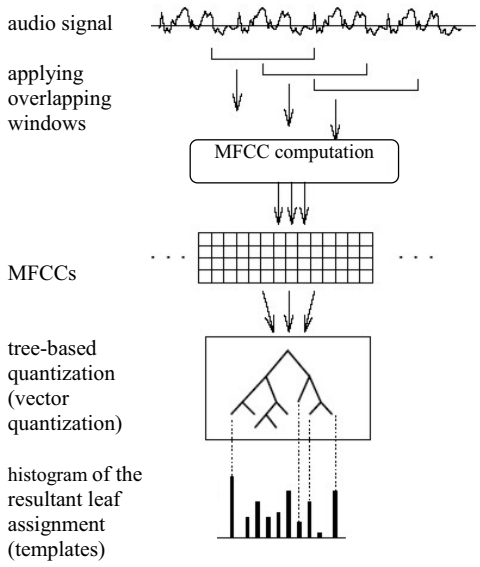


Fig. 4.10. TreeQ system structure (Foote 1997)

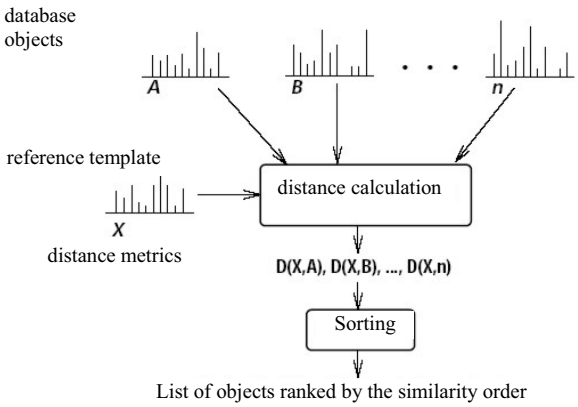


Fig. 4.11. Classification of audio signals based on template histograms (Foote 1997)

The TuneServer mentioned before is interesting from the implementation point of view. The Parson's code was used, which takes into account only the melody direction and discards the rhythm information. This code is specifically resilient to the rhythmic-based errors. The query is based on whistling, and returns the ranked list of results.

Another system MIRACLE (**M**usic **I**nformation **R**etrieval **A**coustically with **C**lustered and paralle**L** **E**ngine), which appeared in 2000, was constructed by Jang, Chen and Kao from Taiwan (http://ismir2001.ismir.net/posters/jang.pdf – MIRACLE). The system is an example of the distributed QBH, but of the *Query-by Singing* type. The system contains 18 slave servers of different processing features. The results are shown as a 'Top 20' list, which is created by either the DTW (*Dynamic Time Warping*) or the HFM method (*two-step hierarchical filtering method*, *two-step HFM*) by comparing the template patterns contained in all slave servers. The main server sorts all lists received and returns the final results.

GUIDO system, engineered by Hoos, Renz and Görg in 2001 is an example of the content-based music information retrieval (http://www.cs.ubc.ca/~hoos/Publ/ismir01.pdf; http://www.informatik.tu-darmstadt.de/AFS/GUIDO). It can be qualified for the *Query-by-Example* systems. The authors of this system developed a GUIDO/XML format, which enables to search the database containing musical scores. The query consists in formulating features characteristic to the type of the searched example. For example, a melody is based on the absolute pitch, pitch-class, interval, interval class, melodic trend (direction of changes, if any: up, down, equal). On the other hand, rhythmical features are as follows: abso-

lute duration of a note, relative duration, rhythmical trend (shorter, longer, equal). The query regards an individual note or a pair of notes, respectively. The engine of the system is based on the Hidden Markov Models. States of the Markov model are related to note features, such as mentioned before, i.e. pitch, intervals, note duration, etc. the database is organized as a tree structure, such a structure supports the system decision taking. In Fig. 4.12 the GUIDO system structure is shown.

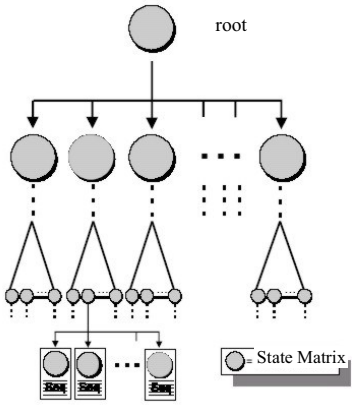


Fig. 4.12. Tree structure of the Guido system (http://www.informatik.tu-darmstadt.de/AFS/GUIDO)

Another system belonging to the content-based music information retrieval is the OMRAS (**O**nline **M**usic **R**etrieval **A**nd **S**earching) project, developed in 2001 in Kings College and the University of Massachusetts (http://omras.dcs.kcl.ac.uk/Full_desc.html). The system supports queries of the MIDI-based files of different types. For example, a query can be formulated as *highly-structured files* (high-level structure of the MIDI files based on music notation with the possibility to distinguish polyphonic lines), *semi-structured files* (medium-level structure of the MIDI files based on a pitch-time matrix, in which '1s' appear starting with the event onset and ending with event release), *unstructured files* – audio files in PCM format. The parameterization method related to the higher level of the MIDI files is based on music indexing, on the other hand, files presented in PCM format use *time–frequency representations*, wavelet representations, etc.

The CubyHum system developed at the University of Eindhoven in 2002 is one of the QBH systems (http://ismir2002.ismir.net/proceedings/02-FP06-2.pdf). A so-called *sub-harmonic summation* technique is used to estimate pitch from the hummed

melody, then event onsets and intervals are determined in the melody. Melodies that are similar to the pitches from the sung melody are retrieved from the database. A dynamic programming framework is used in the pattern matching. The authors of the system say that this application is a "linked combination of speech signal processing, music processing and approximate pattern matching", and in addition, the knowledge derived from the experimental practice.

In 2003 Typke, Veltkamp, and Wiering from the University of Utrecht developed the Orpheus system (http://www.cs.uu.nl/people/rtypke/transdist_ismir2003.pdf), in which the query mechanism principle is based on the Earth Mover's Distance, and its modification. An example from the database, a MIDI file, or a whistled melody can formulate a query. Then, they are converted into the internal database format before the search starts. Note pitch is defined according to the interval-invariant base-40 notation system, proposed by Hewlett. In addition, two types of weights are utilized, namely stress and note number weights. In general their role is to differentiate between note duration and melody structures. The pattern recognition is organized in such a way that a small number of candidates are selected from the database using a KD-tree which is based on Euclidean distances in a space of transportation distances. Then, on the basis of the objects found, a more expensive transportation distance is calculated to obtain the exact distances instead of the distance boundaries.

MelodieSuchmaschine project results from the research of the Fraunhofer Institutes in Erlangen and Ilmenau (http://www.iis.fraunhofer.de/amm/download/qbh.pdf; http://www.cebit2003.fraunhofer.de/servlet/is/4107/). The system was introduced in 2003. The system is of the QBH type. MelodieSuchmaschine exists as an autonomic system or a www application, or a query is introduced by a mobile phone, then processed by the server and returned to the mobile phone. The fundamental frequencies of the hummed melody are transformed to a pitch contour which is subsequently divided into several notes. Each note is characterized by its temporal duration and pitch. The query returns a ranked list of 10 most similar songs. The information on the identified song title, artist, composer, lyrics, etc. is also sent to the user.

There are also some other systems, some of them commercial audio fingerprinting systems, these systems are listed in the MIR society webpage.

4.2.1 Description of Musical Phrase

The analysis of a musical phrase is not a fully solved problem, however it depends both on the quality of a musical phrase representation and on the inference engine.

There are two main types of musical files – audio signals and structured files such as for example MIDI. The unsolved problem in a signal-domain is to extract individual sounds from a stream of audio. Melody contours can be extracted from an audio file by means of various techniques described in the previous Chapter, but chords or sounds playing at the same time cannot be effectively extracted, yet. In the case of MIDI files such a problem does not exist because each note is described as a set of its physical attributes – melodic and rhythmic values, thus the high level analysis is possible. However, there exist elements of a piece, which cannot be easily extracted neither from the audio signal nor from the MIDI code, they concern emotional features for example, and are often described in a non-formal way, e.g. as a description of a mood of a musical piece.

As already mentioned, musicologists list a few elements of a musical piece. For example, Byrd and Crawford (2002) claim that the most informative are melody and rhythm, assigning about 50% of informativeness to a melody, 40% to rhythm and remaining 10% to the elements such as harmony, dynamics, articulation, etc. Since melody is the most important element of a musical piece, thus in this Chapter the experiments presented are concentrated around this feature, however examples of rhythm retrieval techniques are also shown.

The musical-phrase analysis case-studies presented were performed formerly by the author (Kostek 1995, 1998, 1999; Kostek and Szczerba 1996a, 1996b), and also by her colleague Czyzewski, and a Ph.D. student of his, Szczerba (Czyzewski and Szczerba 2002; Czyzewski et al 2004; Szczerba 1999, 2002). This Chapter also comprises a description of another ongoing project carried out by the author and her Ph.D. student, Wojcik (Kostek and Wojcik 2004, 2005; Wojcik and Kostek 2004). Part of the research presented was published in the first volume of Transactions on Rough Sets, printed by Springer Verlag (Czyzewski et al 2004, Czyzewski and Kostek 2004; Kostek and Czyzewski 2004).

The experiments assume the discussed musical phrases to be single-voice ones. This means that at moment t, at most one musical event occurs in the phrase. In general, a musical event is defined as a single sound of the defined pitch, amplitude, duration, onset and timbre (Tanguiane 1991; The New Grove Dictionary). A musical pause — the absence of sound — is a musical event as well. For practical reasons a musical pause was assumed

to be a musical event of pitch equal to the pitch of the preceding sound and of a zero amplitude.

A single-voice musical phrase *fr* can be expressed as a sequence of musical events:

$$fr = \{e_1, e_2, ..., e_n\} \tag{4.32}$$

Musical event e_i can be described as a pair of values denoting sound pitch h_i (in the case of a pause, pitch of the previous sound), and sound duration t_i:

$$e_i = \{h_i, t_i\} \tag{4.33}$$

One can therefore express a musical phrase by a sequence of pitches being a function of time $fr(t)$. A sample illustration of the function $fr(t)$ is presented in Fig. 4.13. Sound pitch is defined according to the MIDI standard, i.e. as a difference from the C0 sound measured in semitones (Braut 1994).

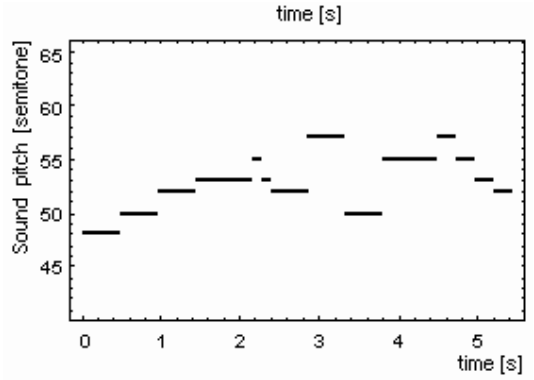


Fig. 4.13. Musical phrase as a function of sound pitch in time. Sound pitch is expressed according to the MIDI standard

One of the basic composer and performer's tools is transforming musical phrases according to rules specific to music perception, and aesthetic and cultural conventions and constraints (Barucha and Todd 1991). Generally, listeners perceive a modified musical phrase as identical to the unmodified original. Modifications of musical phrases involve sound pitch shifting (transposition), time changes (e.g. augmentation), changes of ornament and/or transposition, shifting pitches of individual sounds etc. (Todd 1991). A formal definition of such modifications may be presented by the example of a transposed musical phrase, expressed as follows:

For example, a transposed musical phrase can be expressed as follows:

$$fr_{mod}(t) = fr_{ref}(t) + c \tag{4.34}$$

where $fr_{ref}(t)$ denotes an unmodified, original musical phrase, $fr_{mod}(t)$ is a modified musical phrase, c is a coefficient expressing in semitones the shift of individual sounds of the phrase by a constant factor (for $|c|=12n$ there is an octave shift).

A musical phrase with a changed tempo can be expressed as follows:

$$fr_{mod}(t) = fr_{ref}(kt) \tag{4.35}$$

where k denotes the tempo change coefficient.

A phrase tempo is slowed down for the values of coefficient $k < 1$. Tempo increase is obtained for the values of coefficient $k > 1$. A transposed musical phrase with a changed tempo can be expressed as follows:

$$fr_{mod}(t) = fr_{ref}(kt) + c \tag{4.36}$$

An example of a musical phrase transposition and a tempo change is presented in Fig. 4.14.

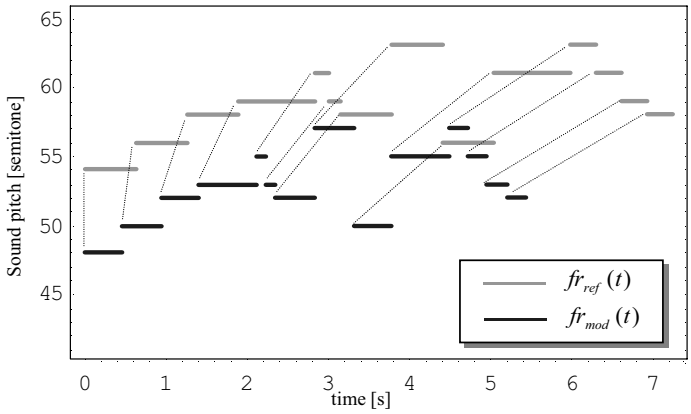


Fig. 4.14. Musical phrase transposition and tempo change ($k = \frac{3}{4}$; $c = 6$)

A variation of tempo in time – a tempo fluctuation – can result mostly from performer's expression or performance inexactness (Desain and Honing 1991, 1997; Repp 1996). Tempo fluctuations can be expressed as function $\Delta k(t)$. A musical phrase with a fluctuating tempo can be expressed as follows:

$$fr_{mod}(t) = fr_{ref}[t\ \Delta k(t)] \tag{4.37}$$

Many methods of modifying melodic content of musical phrases are available. Among the most often used ones are: ornament, transposition, inversion, retrograde, scale change (major – minor), change of pitch of individual sounds (e.g. harmonic adjustment), etc. The above methods can be expressed as melodic modification function $\psi(t)$. Therefore, a musical phrase with melodic content modifications can be expressed as follows:

$$fr_{mod}(t) = fr_{ref}(t) + \psi(t) \tag{4.38}$$

A musical phrase modified by the above means (transposition, a tempo change, a tempo fluctuation and a melodic content modification) can be expressed as follows:

$$fr_{mod}(t) = fr_{ref}\left[kt + t\,\Delta k(t)\right] + \psi\left[kt + t\,\Delta k(t)\right] + c \tag{4.39}$$

Formalizing musical phrase modifications allows for defining the problem of an automatic classification of musical phrases. Let fr_{mod} be a modified or unmodified musical phrase being classified and let FR be a set of unmodified reference phrases:

$$FR = \left\{fr_{1ref}, fr_{2ref}, ..., fr_{Nref}\right\} \tag{4.40}$$

The task of recognizing musical phrase fr_{mod} can therefore be described as finding, in set FR, such a phrase fr_{nref}, for which the musical phrase modification equation is true.

If the utilized modification means are limited to transposition and uniform tempo changes, a modification can be described using two constants: a transposition constant c and a tempo change constant k. In the discussed case the task of classifying a musical phrase is limited to determining such vales of constants c and k that the equation is true. If function $\Delta k(t) \neq 0$, then the mechanism of classification should minimize the influence of the function. Small values of function $\Delta k(t)$ indicate slight changes resulting from articulation inexactness and moderate performer's expression (Desain 1992). Such changes can be corrected by means of time quantization. Larger values of function $\Delta k(t)$ indicate major temporal fluctuations resulting chiefly from the performer's expression. Such changes can be corrected using advanced methods of time quantization (Desain and Honing 1991).

Function $\psi(t)$ describes a wide range of musical phrase modifications that are characteristic for a composer epoch as well as for the performer's style and technique. Values of function $\psi(t)$, which describes qualitatively the character of the above factors, are difficult or impossible to determine in a hard-defined manner. This is the main problem of the automatic classification of musical phrases.

4.2.2 Parametrization of Musical Phrases

Since the subject of parametrization was thoroughly discussed in Chapter 3, thus here only a few remarks will be presented. A fundamental quality of intelligent decision algorithms is their ability to classify data that cannot be exactly defined and modeled mathematically. This quality enables to use intelligent decision algorithms for an automatic classification of musical phrases in the conditions of the lack of a definition and a quality character of $\psi(t)$ and $\Delta k(t)$ functions.

The data to be classified by intelligent decision algorithms can be represented as a vector:

$$\mathbf{P} = \left[p_1, p_2, ..., p_N\right] \tag{4.41}$$

The constant number N of elements of vector **P** requires musical phrase *fr* to be represented by N parameters, independent of the duration of the number of tones in phrase *fr*. Converting a musical phrase *fr* of the shape of $\{e_1, e_2, \ldots, e_n\}$ into an N-element vector of parameters enables the representation of distinctive qualities of musical phrase *fr*. As shown above, the transposition of a musical phrase and the uniform proportional tempo change can be represented as the alteration of two constants: c and k. It would therefore be advantageous to design such method of musical phrase parameterization, for which:

$$\mathbf{P}(fr_{mod}) = \mathbf{P}(fr_{ref}) \tag{4.42}$$

where:

$$fr_{mod}(t) = fr_{ref}(kt) + c \tag{4.43}$$

Creating a numerical representation of musical structures to be used in automatic classification and prediction systems requires defining the following characteristics:

- a sequence size,
- a method of representing sound pitch,
- a method of representing time-scale and frequency properties,
- methods of representing other musical properties by feature vectors.

In addition, after defining various subsets of features, a feature selection should be performed. Typically, this process consists in finding an optimal feature subset from a whole original feature set, which guarantees the accomplishment of a processing goal while minimizing a defined feature selection criterion (Chmielewski and Grzymala Busse 1994; Skowron and Nguyen 1995; Nguyen 1998; Swiniarski 2001). Feature relevance may be

evaluated on the basis of open-loop or closed-loop methods. In the first approach separability criteria are used. To this end the Fisher criterion is often employed. The closed-loop methods are based on feature selection using a predictor performance. This means that the feedback from the predictor quality is used for the feature selection process. On the other hand, this is a situation, in which a feature set contains several disjoint feature subsets. The feature selection defined for the purpose of this study consists in eliminating the less effective method of parametrization according to the processing goal first, and then in reducing the number of parameters to the optimal one. Both, open- and closed-loop methods were used in the study performed.

Individual musical structures may show significant differences in the number of elements, i.e. sounds or other musical units. In the extreme case the classifier should be fed with the whole musical structure (e.g. the melody or the whole musical piece). It is therefore necessary to limit the number of elements in the numerical representation vector. This can be achieved by employing a movable time window of a constant size. Time windows can overlap.

Sound pitch can be expressed as an absolute or a relative value. An *absolute representation* is characterized by the exact definition of a reference point (e.g. the C1 sound). In the case of an absolute representation the number of possible values defining a given sound in a sequence is equal to the number of possible sound pitch values. A disadvantage of this representation is the fact of shifting the values for the sequence elements by a constant factor in the case of transposition. In the case of a *relative representation* the reference point is being updated all the time. Here the reference point may be e.g. the previous sound, a sound at the previously accented time part or a sound at the time start. The number of possible values defining a given sound in a sequence is equal to the number of possible intervals. An advantage of a relative representation is the absence of change in musical structures caused by transposition (i.e. shifting the structure a defined interval up or down) as well as the ability to limit the scope of available intervals without limiting the available musical scales. Its disadvantage is sensitivity to small structure modifications resulting in shifting the reference point.

Research performed so far resulted in designing a number of parametric representations of musical phrases. Some of these methods were described in detail in earlier publications (Kostek 1995, 1998, 1999; Kostek and Szczerba 1996a, 1996b), therefore only their brief characteristics are given below.

Statistical parametrization

The designed statistical parameterization approach is aimed at describing structural features of a musical phrase. The statistical parametric description has been used in previous case studies (Kostek 1995; Kostek and Szczerba 1996a), and introduced parameters are as follows:

- P_1 – the difference between weighted average sound pitch and pitch of the lowest sound of a phrase:

$$P_1 = \left[\frac{1}{T}\sum_{n=1}^{N} h_n t_n\right] - \min_n(h_n) \tag{4.44}$$

where T is the phrase duration, h_n denotes pitch of nth sound, t_n is a duration of nth sound, N denotes the number of sounds in a phrase.

- P_2 – *ambitus* – the difference between the pitches of the highest and the lowest sounds of a phrase:

$$P_2 = \max_n(h_n) - \min_n(h_n) \tag{4.45}$$

- P_3 – the average absolute difference of pitches of subsequent sounds:

$$P_3 = \frac{1}{N-1}\sum_{n=1}^{N-1}\left|h_n - h_{n+1}\right| \tag{4.46}$$

- P_4 – the duration of the longest sound of a phrase:

$$P_4 = \max_n(t_n) \tag{4.47}$$

- P_5 – average sound duration:

$$P_5 = \frac{1}{N}\sum_{n=1}^{N} t_n \tag{4.48}$$

Statistical parameters representing a musical phrase can be divided into two groups: parameters describing melodic quantities of a musical phrase (P_1, P_2, P_3) and parameters describing rhythmical quantities of a musical phrase (P_4, P_5).

An example of the values of statistical parameters calculated for a musical phrase shown in Fig. 4.15 is contained in Table 4.11.

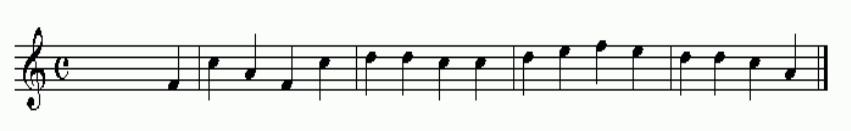


Fig. 4.15. An example of a musical phrase

Table 4.11. Values of statistical parameters calculated for musical phrase shown in Fig. 4.15

Statistical parameter	Parameter value
P_1	7.1764
P_2	12
P_3	2.375
P_4	1024
P_5	1024

Trigonometric Parametrization

Trigonometric parametrization involves representing the shape of a musical phrase with a vector of parameters **P**=[p_1, p_2, …, p_M] of a series of cosines of the form (Kostek and Szczerba 1996b):

$$fr^*(t) = p_1 \cos\left[\left(t-\frac{1}{2}\right)\frac{\pi}{T}\right] + p_2 \cos\left[2\left(t-\frac{1}{2}\right)\frac{\pi}{T}\right] + ... + p_M \cos\left[M\left(t-\frac{1}{2}\right)\frac{\pi}{T}\right] \quad (4.49)$$

For a discrete time-domain, it is assumed that the sampling period is a common denominator of durations of all rhythmic units of a musical phrase. Elements p_m of the trigonometric parameter vector **P** are calculated according to the following formula:

$$p_m = \sum_{k=1}^{l} h_k \cos[m(k-\frac{1}{2})\frac{\pi}{l}] \quad (4.50)$$

where $l = \frac{T}{snt}$ denotes the phrase length expressed as a multiple of a sampling period, *snt* is the shortest note duration, h_k is pitch of sound in *k*th sample.

According to the taken assumption each rhythmic value, which is a multiple of the sampling period is transformed into a series of rhythmic values equal to the sampling period. This leads to the loss of information on the rhythmic structure of the phrase. Absolute changes of values concerning

sound pitch and proportional time changes do not affect the values of trigonometric parameters.

Trigonometric parameters enable to reconstruct the shape of the musical phrase they represent. Phrase shape is reconstructed using vector $\mathbf{K}=[k_1, k_2, ..., k_N]$. Elements of vector **K** are calculated according to the following formula:

$$k_n = \frac{1}{N}\sum_{m=1}^{M} 2P_m \cos\left(\frac{mn\pi}{N}\right) \tag{4.51}$$

where M is the number of trigonometric parameters representing the musical phrase, P_m denotes mth element of the parameters vector.

Values of elements k_n express in semitones the difference between the current and the average sound pitch in the musical phrase being reconstructed.

An example of the values of trigonometric parameters calculated for the musical phrase shown in Fig. 4.15 is contained in Table 4.12.

Table 4.12. Values of trigonometric parameters calculated for the musical phrase shown in Fig. 4.15

Trigonometric parameter	Parameter value
1	-0.9769
2	-1
3	0.3452
4	-0.4971
5	0.2809
6	0.3021
7	0.0354
8	-0.5383
9	-0.3443
10	-0.4899
11	-0.2535
12	-0.2027
13	-0.0920
14	0.0603
15	0.0665

It is interesting to analyze the results of an experiment that aimed at finding the most important features. For such an analysis, a genetic algorithm was used. Feature vectors consisted of 15 trigonometric and 5 statistical parameters. Tests were carried out on the basis of a MIDI database. Approximately 200 excerpts from J.S. Bach's works, such as cantatas, mo-

tets, fugues, chorales, masses and hymns were randomly chosen for the experiment. All phrases have a number assigned (BWV - *Bach Werke Verzeichnis*) according to the www.jsbach.org webpage. Modifications have also been applied to the pattern phrases, such as ornaments, note omission, errors, prolongation, etc. For example, applying six modifications to the patterns results in 1200 phrases used in the training stage.

Fig. 4.16 shows the fitness process during the training phase (distance minimization). One can observe that after some populations have been produced, the optimization process achieves an asymptotic value, which is not exceeded in subsequent populations. In Fig. 4.17 results of the genetic algorithm performance are shown.

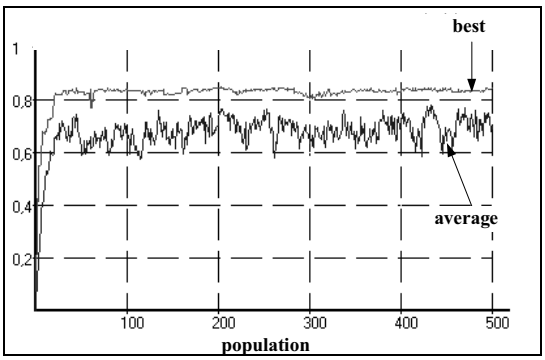


Fig. 4.16. Adaptation process in subsequent populations during the training process

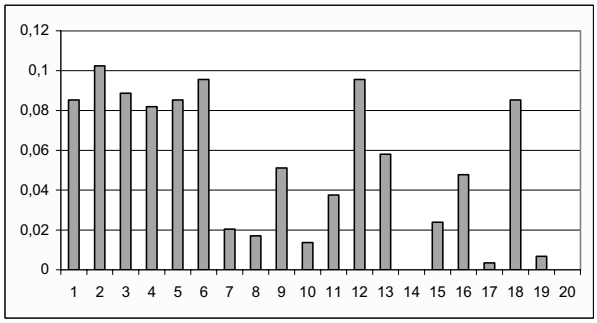


Fig. 4.17. Optimum weights applicable to trigonometric and statistical parameters (first 15 parameters concern trigonometric parameters, the following features are statistical ones)

As seen in Fig. 4.17 parameters differ in significance, therefore it would be desirable to apply such a process of optimization of parameter weights

based on genetic algorithms before the main experiments on classification are performed, especially as this would allow for eliminating redundant features.

Polynomial Parametrization

Single-voice musical phrase *fr* can be represented by function *fr*(*t*), whose domain is time-interpreted either discretely or as a continuum. In a discrete time-domain musical phrase *fr* can be represented as a set of points in a two-dimensional space of time and sound pitch. A musical phrase can be represented in a discrete time-domain by points denoting sound pitch at time *t*, or by points denoting note starts.

If a tempo varies in time (function $\Delta k(t) \neq 0$) or a musical phrase includes additional sounds of durations inconsistent with the general rhythmic order (e.g. ornament or augmentation), a sampling period can be determined by minimizing the quantization error defined by the formula:

$$\varepsilon(b) = \frac{1}{N-1} \sum_{i=1}^{N-1} \left| \frac{t_i - t_{i-1}}{b} - Round\left(\frac{t_i - t_{i-1}}{b} \right) \right| \tag{4.52}$$

where *b* is a sampling period, and *Round* is a rounding function.

On the basis of the representation of a musical phrase in a discrete-time domain one can rough the representation points by a polynomial of the following form:

$$fr^*(t) = a_0 + a_1 t + a_2 t^2 + \ldots a_M t^M \tag{4.53}$$

Coefficients $a_0, a_1, \ldots a_M$ are found numerically by means of mean-square approximation, i.e. by minimizing the error ε of form:

$$\varepsilon^2 = \int_0^T \left| fr^*(t) - fr(t) \right|^2 dt \quad - \text{ continuous case} \tag{4.54}$$

$$\varepsilon^2 = \sum_{i=0}^{N} \left| fr^*{}_i - fr_i \right|^2 \quad - \text{ discrete case}$$

One can also express the error in semitones per sample, which facilitates the evaluation of approximation, according to the formula:

$$\chi = \frac{1}{N} \sum_{i=1}^{N} \left| fr_i^* - fr_i \right| \tag{4.55}$$

Binary Parametrization

Binary representation is based on dividing the time window W into n equal time sections T, where n is consistent with a metric division and T corresponds to the smallest basic rhythmic unit in the music material being represented. Each time section T is assigned a bit of information b_T in the vector of rhythmic units. Bit b_T takes the value of 1, if a sound begins in the given time section T. If time section T covers a sound started in a previous section or a pause, the rhythmic information bit b_T assumes the value of 0.

An advantage of a binary representation of rhythmic structures is the constant length of a sequence representation vector. Its disadvantages are: large vector length in comparison with other representation methods and the possibility of errors resulting from time quantization.

On the basis of methods used in representing the values of individual musical parameters one can distinguish three types of representations: local, distributed and global ones.

In the case of a *local representation* every musical unit e_n is represented by a vector of n bits, where n is the number of all possible values of a musical unit e_n. The current value of a musical unit e_n is represented by ascribing the value of 1 to the bit of the representation vector corresponding to this value. Other bits of the representation vector take the value of 0 (unipolar activation) or –1 (bipolar activation). This type of representation was used e.g. by Hörnel (1997) and Todd (1991).

The system of representing musical sounds proposed by Hörnel and his co-worker is an example of a parametric representation (Feulner and Hörnel 1994). In this system each subsequent note p is represented by the following parameters:

- consonance of note p with respect to its harmony,
- a relation of note p towards its successor and predecessor in the case of dissonance against the harmonic content,
- a direction of p (up, down to next pitch),
- a distance of note p to base note (if p is consonant),
- an octave,
- *tenuto* – if p is an extension of the previous note of the same pitch.

The presented method of coding does not employ a direct representation of sound pitch; it is distributed with respect to pitch. Sound pitch is coded as a function of harmony.

In the case of a *distributed representation* the value of musical unit E is encoded with m bits according to the formula:

$$m = \lfloor \log_2 N \rfloor \tag{4.56}$$

where:

N – the number of possible values of musical unit e_n.

A distributed representation was used e.g. by Mozer (1991). An example of representing the sounds of the chromatic scale using a distributed representation is presented in Table 4.13.

Table 4.13. Distributed representation of sound pitches according to Mozer

Sound pitch	Mozer's distributed representation					
C	–1	–1	–1	–1	–1	–1
$C^{\#}$	–1	–1	–1	–1	–1	+1
D	–1	–1	–1	–1	+1	+1
$D^{\#}$	–1	–1	–1	+1	+1	+1
E	–1	–1	+1	+1	+1	+1
F	–1	+1	+1	+1	+1	+1
$F^{\#}$	+1	+1	+1	+1	+1	+1
G	+1	+1	+1	+1	+1	–1
$G^{\#}$	+1	+1	+1	+1	–1	–1
A	+1	+1	+1	–1	–1	–1
$A^{\#}$	+1	+1	–1	–1	–1	–1
B	+1	–1	–1	–1	–1	–1

In the case of a *global representation* the value of a musical unit is represented by a real value.

The above methods of representing values of individual musical units imply their suitability for processing certain types of music material, for certain tasks and analysis tools, classifiers and predictors.

Prediction of Musical Events

The experiments were aimed at designing a method of predicting and entropy-coding of music. A concept of entropy-coding was presented by Shannon and later used e.g. for investigating the entropy of texts in English by Moradi, Grzymala–Busse and Roberts (1998). The engineered method was used as a musical event predictor in order to enhance a system of pitch detection of a musical sound.

The block scheme of a prediction coding system for music is presented in Fig. 4.18.

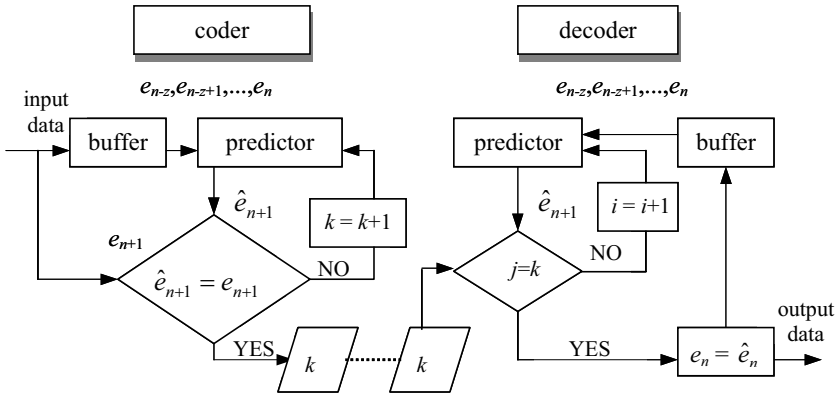


Fig. 4.18. Block scheme of prediction coder and decoder

The idea of entropy coding involves using two identical predictors in the modules of data coding and decoding. The process of coding involves determining the number of prediction attempts k required for correct prediction of event e_{n+1}. Prediction is based on parameters of musical events collected in data buffer. The number of prediction attempts k is sent to the decoder. The decoder module determines the value of event e_{n+1} by repeating k prediction attempts. Subsequent values for samples – musical events – are then collected in a buffer.

Two types of data buffers were implemented:

- data buffer of limited capacity,
- fading memory model.

In the case of a data buffer of limited capacity the buffer stores data on z musical events; each event is represented by a separate vector. That means that z vectors representing z individual musical events are supplied to the input of a predictor. During the research the value set (z) was limited to 5, 10 and 20 samples.

The fading memory model involves storing and fading the previous values of the vector elements in the buffer and summing them with the current values according to the formula:

$$b_n = \sum_{k=1}^{n} e_k r^{n-k} \tag{4.57}$$

where r is a fading factor from the range (0,1).

In the case of using the fading memory model a single vector of parameters of musical events is supplied to the input of a predictor. This means a

z-fold reduction of the number of input parameters compared with the buffer of size of *z*.

For the needs of investigating the music predictor a set of diversified musical data representing melodic data (concerning pitch of subsequent sounds) and rhythmic data (concerning relative durations of subsequent musical events) was prepared. The experiment was based on a system of musical data prediction utilizing artificial neural networks. First, a series of experiments aimed at optimizing the predictor structure, data buffer parameters and prediction algorithm parameters were performed.

In the process of training the neural musical predictor all voices of the individual fugues except from the highest ones were utilized. The highest voices were used for testing the predictor. Three methods of a parametric representation of sound pitch: binary method, a so-called modified Hörnel's representation and a modified Mozer's representation were utilized. In all cases a relative representation was used, i.e. differences between pitch of subsequent sounds were coded.

In the case of a binary representation individual musical intervals (differences between pitch of subsequent sounds) are represented as 27-bit vectors. The utilized representation of sound pitch is presented in Fig. 4.19.

-octave	Interval [in semitones]																								+octave
	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11	+12	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0

Fig. 4.19. Illustration of a binary representation of a musical interval (example – 2 semitones up)

The Hörnel's representation of intervals discussed in this study is a diatonic representation (corresponding to seven-step musical scale). For the needs of this research study a modified Hörnel's representation that enables chromatic (twelve-step) representation was used. Individual intervals are represented by means of 11 parameters. A method employed for coding intervals is presented in Table 4.14.

Table 4.14. Modified Hörnel's representation of intervals

Interval [in semi-tones]	Direction bits	Octave bit	Interval size representation parameters						
-13	1 0 0	1	0.5	0	0	0	0	0	1
-12	1 0 0	1	1	0	0	0	0	0	0.5
-11	1 0 0	0	1	0.5	0	0	0	0	0
-10	1 0 0	0	0.5	1	0	0	0	0	0
-9	1 0 0	0	0	1	0.5	0	0	0	0
-8	1 0 0	0	0	0.5	1	0	0	0	0
-7	1 0 0	0	0	0	0.5	1	0	0	0
-6	1 0 0	0	0	0	0	1	0.5	0	0
-5	1 0 0	0	0	0	0	0.5	1	0	0
-4	1 0 0	0	0	0	0	0	1	0.5	0
-3	1 0 0	0	0	0	0	0	0.5	1	0
-2	1 0 0	0	0	0	0	0	0	1	0.5
-1	1 0 0	0	0	0	0	0	0	0.5	1
0	0 1 0	0	1	0	0	0	0	0	0.5
+1	0 0 1	0	1	0.5	0	0	0	0	0
+2	0 0 1	0	0.5	1	0	0	0	0	0
+3	0 0 1	0	0	1	0.5	0	0	0	0
+4	0 0 1	0	0	0.5	1	0	0	0	0
+5	0 0 1	0	0	0	0.5	1	0	0	0
+6	0 0 1	0	0	0	0	1	0.5	0	0
+7	0 0 1	0	0	0	0	0.5	1	0	0
+8	0 0 1	0	0	0	0	0	1	0.5	0
+9	0 0 1	0	0	0	0	0	0.5	1	0
+10	0 0 1	0	0	0	0	0	0	1	0.5
+11	0 0 1	0	0	0	0	0	0	0.5	1
+12	0 0 1	1	1	0	0	0	0	0	0.5
+13	0 0 1	1	1	0.5	0	0	0	0	0

The distribution method of representing sound pitch designed by Mozer is an absolute representation method. Within the scope of this research a modified Mozer's representation was introduced by Szczerba. The representation enables a relative representation of the pitch of subsequent sounds (i.e. for representing intervals). It was complemented by adding a direction parameter and an octave bit. An individual musical event is therefore coded by means of 8 parameters. The employed method of coding intervals is presented in Table 4.15.

Table 4.15. Modified Mozer's representation of intervals

Interval [in semitones]	Direction bit	Octave bit	Interval size representation parameters					
-13	-1	1	+1	–1	–1	–1	–1	–1
-12	-1	1	–1	–1	–1	–1	–1	–1
-11	-1	0	–1	–1	–1	–1	–1	+1
-10	-1	0	–1	–1	–1	–1	+1	+1
-9	-1	0	–1	–1	–1	+1	+1	+1
-8	-1	0	–1	–1	+1	+1	+1	+1
-7	-1	0	–1	+1	+1	+1	+1	+1
-6	-1	0	+1	+1	+1	+1	+1	+1
-5	-1	0	+1	+1	+1	+1	+1	–1
-4	-1	0	+1	+1	+1	+1	–1	–1
-3	-1	0	+1	+1	+1	–1	–1	–1
-2	-1	0	+1	+1	–1	–1	–1	–1
-1	-1	0	+1	–1	–1	–1	–1	–1
0	0	0	–1	–1	–1	–1	–1	–1
+1	+1	0	–1	–1	–1	–1	–1	+1
+2	+1	0	–1	–1	–1	–1	+1	+1
+3	+1	0	–1	–1	–1	+1	+1	+1
+4	+1	0	–1	–1	+1	+1	+1	+1
+5	+1	0	–1	+1	+1	+1	+1	+1
+6	+1	0	+1	+1	+1	+1	+1	+1
+7	+1	0	+1	+1	+1	+1	+1	–1
+8	+1	0	+1	+1	+1	+1	–1	–1
+9	+1	0	+1	+1	+1	–1	–1	–1
+10	+1	0	+1	+1	–1	–1	–1	–1
+11	+1	0	+1	–1	–1	–1	–1	–1
+12	+1	1	–1	–1	–1	–1	–1	–1
+13	+1	1	–1	–1	–1	–1	–1	+1

A relative binary representation was designed for coding rhythmic values. Rhythmic values are coded by a feature vector:

$$\mathbf{p}^r = \left\{p_1^r, p_2^r, p_3^r, p_4^r, p_5^r\right\} \tag{4.58}$$

where individual parameters assume the values:

$$p_1^r = \frac{\left|\frac{e_{n-1}^r}{e_n^r} - 2\right| + \frac{e_{n-1}^r}{e_n^r} - 2}{12} \tag{4.59}$$

$$p_2^r = \begin{cases} \dfrac{\left|8 - \dfrac{e_{n-1}^r}{e_n^r}\right| + 8 - \dfrac{e_{n-1}^r}{e_n^r}}{12} & \text{for} \quad \dfrac{e_{n-1}^r}{e_n^r} \geq 2 \\ \dfrac{\left|\dfrac{e_{n-1}^r}{e_n^r} - 1\right| + \dfrac{e_{n-1}^r}{e_n^r} - 1}{2} & \text{for} \quad \dfrac{e_{n-1}^r}{e_n^r} < 2 \end{cases} \tag{4.60}$$

$$p_3^r = \begin{cases} \dfrac{\left|2 - \dfrac{e_n^r}{e_{n-1}^r}\right| + \left(2 - \dfrac{e_n^r}{e_{n-1}^r}\right)}{2} & \text{for} \quad \dfrac{e_n^r}{e_{n-1}^r} \geq 1 \\ \dfrac{\left|2 - \dfrac{e_{n-1}^r}{e_n^r}\right| + \left(2 - \dfrac{e_{n-1}^r}{e_n^r}\right)}{2} & \text{for} \quad \dfrac{e_n^r}{e_{n-1}^r} < 1 \end{cases} \tag{4.61}$$

$$p_4^r = \begin{cases} \dfrac{\left|8 - \dfrac{e_n^r}{e_{n-1}^r}\right| + 8 - \dfrac{e_n^r}{e_{n-1}^r}}{12} & \text{for} \quad \dfrac{e_n^r}{e_{n-1}^r} \geq 2 \\ \dfrac{\left|\dfrac{e_n^r}{e_{n-1}^r} - 1\right| + \dfrac{e_n^r}{e_{n-1}^r} - 1}{2} & \text{for} \quad \dfrac{e_n^r}{e_{n-1}^r} < 2 \end{cases} \tag{4.62}$$

$$p_5^r = \frac{\left|\dfrac{e_n^r}{e_{n-1}^r} - 2\right| + \dfrac{e_n^r}{e_{n-1}^r} - 2}{12} \tag{4.63}$$

where:

e_n^r – a rhythmic value (duration) of musical event e_n.

Values of parameters of rhythmic representation $\mathbf{p}_{,}^r$ dependent on the $\dfrac{e_n^r}{e_{n-1}^r}$ ratio are presented in Table 4.16.

Table 4.16. Relative representation of rhythmic values

$\frac{e_n^r}{e_{n-1}^r}$	p_1^r	p_2^r	p_3^r	p_4^r	p_5^r
⅛	1	0	0	0	0
¼	0.5	0.5	0	0	0
½	0	1	0	0	0
1	0	0	1	0	0
2	0	0	0	1	0
4	0	0	0	0.5	0.5
8	0	0	0	0	1

Presented methods of representing sound pitch and rhythmic values were used to prepare a set of data for investigating the neural musical predictor. Specifications of data sets are presented in Table 4.17.

Table 4.17. Specifications of musical data used for investigating the neural musical predictor (where: mbin_rn denotes a data set containing binary and relative representation, etc.)

Database indicator	pitch representation		time representation		total number of parameters per sample
	representation	parameters/sample	representation	parameters/sample	
mbin_rn	relative - binary	27	NO	0	27
mhor_rn	modified Hörnel	11	NO	0	11
mmoz_rn	modified Mozer	8	NO	0	8
mbin_rrel	relative - binary	27	relative	5	32
mhor_rrel	modified Hörnel	11	relative	5	16
mhor_rrel	modified Mozer	8	relative	5	13

At the first stage of investigating a buffer of a constant size of 5, 10 and 20 samples, respectively, and the values of parameter r for the fading memory model from the set r={0.2; 0.5; 0.8} were chosen.

4.2.3 Neural Musical Predictor

The neural musical predictor was implemented using the *Stuttgart Neural Network Simulator* (SNNS) integrated system for emulating artificial neural networks (Zell 2002). Experiments were performed for individual data

sets presented in Table 4.17 and for various data buffer parameters. Musical material for this study was based on fugues from the set of *Well-Tempered Clavier* by J. S. Bach. The experiments were divided into two stages:

- Investigating the predictor for individual methods of representing data and buffer parameters for data limited to 6 fugues chosen at random (no. 1, 5, 6, 8, 15 and 17); the number of training elements, depending on buffer size, ranged from 5038 to 5318 samples, while the number of test elements ranged from 2105 to 2225 samples, respectively.
- Investigating chosen representation methods and buffer parameters for all 48 fugues from the "*Well-Tempered Clavier*" collection.

The description of the developed predictor and its training process as well as the obtained results of musical data prediction are presented below.

A *feed-forward* neural network model with a single hidden layer. In the cases of a binary representation and a modified Hörnel's representation, a unipolar, sigmoidal shaping function was used, while in the case of a modified Mozer's representation, a hiperbolic tangent bipolar function was used. The choice of the ANN activation function was determined on the basis of pilot tests performed before the main experiments started.

In the first stage of this research the number of neurons in the hidden layer was arbitrarily limited to the set {20, 50, 100}. A series of the cycles of training the neural predictor for individual methods of representing musical events and data buffer parameters was conducted. Due to practical considerations the number of iterations was arbitrarily limited to 1000. The error back-propagation algorithm augmented by the *momentum* method was applied. On the basis of pilot tests, constant values of training process coefficient η=0.5 and the *momentum* coefficient α=0.2 were assumed.

In general, in the cases of a binary representation and the modified Hörnel's representation the mean-square error (MSE) value was reached as early as after 100 iterations, independently to what the number of neurons in the hidden layer was. Conversely, in the case of a modified Mozer's representation (without rhythm representation) the training process did not lead to the mean-square error value lower than 1.

In order to evaluate the performance of the neural musical predictor, the following parameters were considered:

- the measure of the first-attempt prediction correctness:

$$pc = \frac{1}{N}\sum_{n=1}^{N} id\left(e_{n+1}, \hat{e}_{n+1}\right) \tag{4.64}$$

where:

e_{n+1} – factual value of an event *n+1*,

$\hat{e}_{n+1}$ – estimated value of an event *n+1*,

$$id\left(e_{n+1}, \hat{e}_{n+1}\right) = \begin{cases} 1 & dla \quad \hat{e}_{n+1} = e_{n+1} \\ 0 & dla \quad \hat{e}_{n+1} \neq e_{n+1} \end{cases} \tag{4.65}$$

- the average number of prediction attempts required for correct estimation of the value of an event *n+1*,
- the bottom and top estimate of entropy of musical data F_N according to:

$$\sum_{i=1}^{M} i\left(q_i^N - q_{i+1}^N\right)\log_2 i \le F_N \le \sum_{i=1}^{M} - q_i^N \log_2 i q_i^N \tag{4.66}$$

where q_i^N is the frequency of correct estimations of the value of an event e_{n+1} for the *i*th prediction attempt, and M denotes number of possible values of an event e_n.

Subsequent prediction attempts and the recognition of an event identity are emulated by ordering the list of possible values of an event e_{n+1} by the distance:

$$o_m = \sum_{i=1}^{N} \left| e_{n+1,i} - e_{m,i} \right| \tag{4.67}$$

where $m = \{1, 2, \ldots, M\}$, i is the representation parameter index, N denotes the number of representation parameters, and e_m is a possible value of an event e_{n+1},

On the basis of the results obtained in the first stage of this investigation the following conclusions concerning operation of the developed predictor can be presented. First of all, a very high effectiveness was obtained for musical data prediction (above 97%) in the cases of binary representations and the modified Hörnel's representations together with a constant-size data buffer, irrespectively of the representation of rhythmic data. In addition, the application of the fading memory model leads to a degradation of prediction effects (max. ca 75%) with a simultaneous reduction of data and computational complexity of the training process. Also, the application of the modified Mozer's method of representation results in a low prediction of effectiveness. What is also important, the method developed for rhythm coding shows high effectiveness of representation and enables obtaining high correctness of rhythm prediction of musical data.

Tests Employing Whole Collection

In the next stage, tests of the music predictor for the whole range of music sets, i.e. for 48 fugues from the *Well-Tempered Clavier* collection were performed. Neural networks trained using six randomly-chosen fugues were used. Tests employed data that have not been used for training. Neural network parameters and obtained results are presented in Table 4.18 (number of neurons in the hidden layer was equal to 50).

Table 4.18. Predictor parameters and prediction results for the whole collection (denotations as previously introduced)

Database indicator	Buffer type	Buffer size	No. of input parameters	No. of output parameters	Prediction effectiveness			Average number of predictions			Upper approx. of entropy		
					melody	rhythm	total	melody	rhythm	total	melody	rhythm	total
mbin_rn	constant-size	10	270	27	0.54	-	0.54	4.12	-	4.12	1.62	-	1.62
mbin_rrel	constant-size	10	320	32	0.6	0.91	0.58	5.21	1.15	8.31	1.44	0.34	1.4
mbin_rn	constant-size (r=0.8)	1	27	27	0.32	-	0.32	8.24	-	8.24	1.92	-	1.92
mbin_rrel	f. coeff. (r=0.8)	1	32	32	0.35	0.84	0.31	6.94	1.42	12.4	1.83	0.65	1.78
mhor_rn	constant-size	10	110	11	0.53	-	0.53	5.27	-	5.27	1.62	-	1.62
mhor_rrel	constant-size	10	160	16	0.57	0.85	0.52	4.76	1.3	9.14	1.49	0.63	1.28
mhor_rn	f. coeff. (r=0.8)	1	11	11	0.28	-	0.28	5.62	-	5.62	1.62	-	1.62
mhor_rrel	f. coeff. (r=0.8)	1	16	16	0.33	0.82	0.3	7.12	1.42	16.4	1.82	0.67	1.51
mmoz_rn	constant-size	10	80	8	0.12	-	0.12	8.56	-	8.56	1.73	-	1.73
mmoz_rrel	constant-size	10	130	13	0.24	0.72	0.19	6.12	1.85	18.4	1.62	0.72	1.51
mmoz_rn	f. coeff. (r=0.8)	1	8	8	0.07	-	0.07	11.2	-	11.2	1.82	-	1.82
mmoz_rrel	f. coeff. (r=0.8)	1	13	13	0.14	0.61	0.11	13.8	3.18	24.6	1.91	0.91	1.46

The obtained results lead to conclusions that, in general, the developed system enables effective prediction of musical data. The highest effectiveness of prediction was obtained for a binary representation of sound pitch and a modified Hörnel's representation. The use of constant-size buffer enables a more effective prediction compared to the fading memory model, and what was perhaps not very surprising that musical data in J. S. Bach's fugues possess low entropy.

4.2.4 Classification of Musical Phrases

Within the scope of the experiments, two methods of classifying musical phrases that use artificial neural networks and rough sets were developed:

- method of classifying phrases on the basis of the sequences of musical data,
- method of classifying phrases on the basis of the parameters of musical data.

Musical phrases can be classified on the basis of the sequences of musical data. Similarly to the prediction of musical events, the classification is based on parameters collected in a data buffer. In order to unify the size of input data the fading memory approach was used. Experiments were performed on the basis of a binary representation of melody and of a modified Hörnel's representation. Tests using a binary rhythm representation were also performed. Repetitions of musical phrases in a data stream were identified in time windows. A window size can be adapted dynamically along a musical piece using the histogram of rhythmic values.

Musical phrases can also be classified on the basis of the parameters of musical phrases. Three methods of a parametric representation, namely statistical, trigonometric and polynomial ones were applied. Like in the case of analysis of musical data sequences, repetitions of musical phrases were identified in time windows of a dynamically determined size.

Application of Artificial Neural Networks

Two types of neural networks were developed for classifying musical phrases:

- classifier, assigning phrases to single objects from the reference phase set,
- neural comparator, analyzing similarities between musical phrases.

Both classifier types are illustrated in Fig. 4.20.

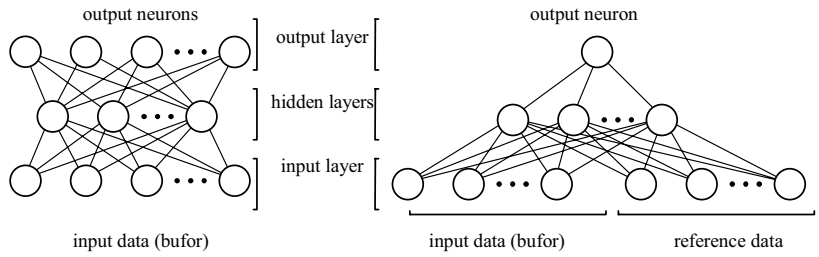


Fig. 4.20. Neural classifier and comparator of musical phrases

In the case of a neural classifier of musical phrases, classification involves determining the phrase identity on the basis of knowledge modeled using the neural network. Individual neurons of the output layer correspond to individual classes of musical phrases. The identity of an input phrase is determined based on the output signals of neurons from the output layer. The classifier scope of application is therefore limited to the scope of reference objects used for network training.

In the case of a neural comparator of musical phrases, classification involves determining the identity relation between the test phrase and the reference phrase. In this case the neural network models the knowledge of the relations between the musical phrase modification and the phrase identity. The comparator scope of application is therefore limited by the scope of phrase modifications used for network training.

Application of the Rough Set-Based Method

The studies also involved the classification of musical phrases using the rough set approach (Pawlak 1982). In this approach both musical data sequences and musical phrase parameters were used in the classification process. Like in the case of classifying musical phrases with the use of artificial neural networks, two systems were developed:

- classifier assigning phrases to single objects from the reference phase set,
- comparator, analyzing similarities between musical phrases.

In this investigation the ROSETTA system developed by the University of Trondheim in cooperation with Warsaw University was employed (Komorowski et al 1998; Øhrm 1999).

4.2.5 Musical Phrase Classification

In order to investigate the methods of classification, sets of musical data were prepared. Fugues from the set of *Well-Tempered Clavier* by J. S. Bach were used as a musical material for this study. Bach's fugues from the *Well Tempered Clavie*r were played on a MIDI keyboard and then transferred to the computer hard disk trough the MIDI card and the Cubase VST 3.5 program. Musical data were grouped in two databases:

- TOFFEE (Themes of Fugues from Well-Tempered Clavier) a musical database containing fugue themes (Kostek 1995),
- DWK (Well-Tempered Clavier) a musical database containing full musical material.

The DWK database was based on MIDI files prepared by Reyto and available from *Classical MIDI Archives* (http://www.prs.net). In order to analyze the classification algorithms in the DWK database, the points of fugue theme repetitions were marked.

The TOFFEE musical database was created in the Department of Multimedia Systems on the basis of fugue themes (Kostek 1998, 1999; Szczerba 1999). Each of the 48 themes was included in the TOFFEE database in the following forms:

- reference form,
- form with a performance error,
- transposed form with a performance error and an ornament,
- transposed form with a performance error,
- form with an additional note at the beginning of the phrase,
- form with the omission of the first note of the phrase + an ornament,
- form with an ornament,
- transposed forms (5),
- augmented form (individual rhythmic values elongated twice),
- transposed augmented form.

The musical phrase modifications that were used, are consistent with the stylistics of music and the technique of performance (Bullivant 1980).

Table 4.19 summarizes methods of representing musical data as well as methods of classifying phrases. The classification methods are denoted with the corresponding symbols:

- ANN – Artificial Neural Networks,
- RS – rough set-based method.

Table 4.19. Methods of representing musical phrases for individual classification methods

Data representation		No. of parameters	Classification algorithm
melodic	rhythmic		
relative, binary	-	27	ANN
relative, Hörnel	-	11	ANN
relative, binary	relative, binary	32	ANN
relative, Hörnel	relative, binary	16	ANN
relative	-	1	RS
relative	relative	2	RS
statistical parameters		5	ANN/RS
trigonometric parameters		5, 10, 15, 20	ANN/RS
polynomial parameters		5, 10, 15	ANN/RS

In the case of non-parametric representations the classification of musical phrases using artificial neural networks and the rough set approach is

based on parameters collected in a data buffer. The fading memory model was utilized. Classifier structure and setting will be presented later on in subsequent paragraphs along with obtained results.

4.2.6 Sequence-Based Classification of Musical Events

Neural Classifier

The research utilizing the neural classifier was divided into two stages:

- investigating the classification in the case of training with reference forms only,
- *leave-one-out* tests for individual musical phrase modifications.

Melodic data were represented using a binary representation, the modified Hörnel's method and the modified Mozer's method. Tests were run for collections both with and without representation of rhythmical data. Rhythm was represented using the relative representation approach. In the case of training performed with reference forms only, it can generally be observed that increasing the number of iterations from 1000 to 2000 resulted in a better effectiveness of classification which rose from ca 1% to the maximum value of 13%. Moreover, it was observed that during the training process the value of a buffer fading coefficient influenced the course of training. For the greatest value of a fading coefficient (0.8), the mean-square error reached the value of 0.1 faster for sets with a rhythm representation (ca 300 iterations) than for sets without a rhythm representation (ca 700 iterations). In the case of a fading coefficient of 0.2 the mean-square error of 0.1 was reached for sets with a rhythm representation after ca 1700 iterations, while this value could not be achieved in the case of sets without a rhythm representation. In general, a higher effectiveness was achieved after taking into account a rhythm representation. In the cases of a binary representation and a modified Hörnel's representation values of over 90% were achieved.

The maximum classification effectiveness (92.63%) was achieved for a binary representation of melodic data with a relative rhythm representation, for a neural network containing 20 neurons in the hidden layer. Table 4.20 summarizes results of classification by ANNs for individual musical phrase representations.

Leave-one-out tests were performed for the modified Hörnel's method of representing melodic data. The effectiveness of classification as a function of a rhythmic data representation was analyzed. For practical reasons

the process of a network training was limited to 1000 iterations. Table 4.21 presents data concerning structures of employed neural networks as well as the information on the classification results.

Table 4.20. Classification results for individual musical phrase representations for neural network containing 20 neurons in the hidden layer (binary representation of melodic data, relative rhythm representation)

Modification	Error	Error + transposition + ornament	Error + transposition	Additional note (beginning of the phrase)	Omission of the first note + ornament	Ornament	Transposition	Augmentation	Augmentation + transposition
Classification effectiveness [%]	87.5	64.58	89.53	95.83	72.92	52.08	97.91	97.91	97.91

Table 4.21. Parameters of neural networks and classification results for leave-one-out tests. Best results for individual musical phrase modifications are marked as bold

Database	Fading coeff. *r*	Number of neurons in hidden layer	Musical phrase modification/ Classification effectiveness [%]									
			No modif.	Error	Error+ transp.+ ornament	Error + transp.	Addit. note	Omission of the first note + ornam.	Ornam.	Transp.	Augm.	Augm. transp.
mhor	0.8	20	100	85.4	68.8	79.2	93.8	85.4	68.8	97.9	97.9	100
mhor	0.8	50	**100**	81.3	79.2	83.3	93.8	**85.4**	70.8	97.9	**100**	**100**
mhor	0.5	20	97.9	87.5	79.2	83.3	95.8	77.1	62.5	95.8	97.9	95.8
mhor	0.5	50	95.8	85.4	81.3	83.3	93.8	83.3	68.8	95.8	97.9	97.9
mhor	0.2	20	91.7	85.4	68.8	62.5	77.1	72.9	64.6	83.3	87.5	95.8
mhor	0.2	50	87.5	79.2	72.9	64.6	77.1	70.8	66.7	91.7	95.8	93.8
mhor_rel	0.8	20	97.9	89.6	72.9	81.3	95.8	**85.4**	58.3	**100**	**100**	**100**
mhor_rel	0.8	50	**100**	89.6	81.3	89.6	93.8	**85.4**	68.8	**100**	**100**	97.9
mhor_rel	0.5	20	**100**	**93.8**	**87.5**	89.6	**97.9**	83.3	64.6	**100**	95.8	**100**
mhor_rel	0.5	50	**100**	91.7	85.4	**91.7**	95.8	**85.4**	66.7	**100**	**100**	**100**
mhor_rel	0.2	20	95.8	87.5	79.2	81.3	87.5	81.3	**72.9**	95.8	93.8	**100**
mhor_rel	0.2	50	95.8	91.7	79.2	81.3	89.6	**85.4**	**72.9**	97.9	95.8	97.9

The maximum classification effectiveness (91.67%) was achieved for data including rhythmic information, buffer fading coefficient r=0.5 and for neural network containing 50 neurons in the hidden layer. The obtained

results confirm the usefulness of representing rhythmic data for proper classification of musical phrases.

Neural Comparator

The classification of musical phrases on the basis of the representation of musical data sequences was investigated using the neural comparator. A set of 12 randomly chosen fugue themes from the TOFFEE database (No. 2, 4, 11, 14, 16, 19, 21, 25, 26, 32, 34, 46) was used for training. The other fugue themes were used for testing the system. The number of iterations during the training process was limited to 2000. Musical data were represented with the use of the modified Hörnel's method and a relative rhythm representation. For each neural network of the comparator the classification effectiveness for the arbitrary classification threshold *th*=0.5 was analyzed. In addition, for each neural network the threshold value was optimized. The effectiveness of a positive classification (a correct identification of a form) and of a negative classification was analyzed. Table 4.22 presents data concerning structures of employed neural networks as well as the information on the obtained results.

Table 4.22. Parameters of neural networks and classification results for neural comparator of musical phrases

			th=0.5		Optimization of the *th* threshold value				
Database indicator	Fading coefficient	Number of neurons in the hidden layer	classification effectiveness positive [%]	classification effectiveness negative [%]	classification effectiveness - average [%]	threshold value of *th*	classification effectiveness positive [%]	classification effectiveness -negative [%]	classification effectiveness - average [%]
mhor_rrel	0.8	20	65.67	98.09	81.88	0.12	75.99	96.52	86.26
mhor_rrel	0.8	50	73.41	97.16	85.29	0.02	80.95	91.44	86.20
mhor_rrel	0.5	20	70.24	97.57	83.90	0.2	82.54	96.13	89.33
mhor_rrel	0.5	50	80.16	98.46	89.31	0.08	90.67	97.09	93.88
mhor_rrel	0.2	20	65.08	96.99	81.03	0.06	77.98	95.53	86.75
mhor_rrel	0.2	50	78.57	95.26	86.92	0.02	94.05	91.46	92.75

4.2.7 Feature Vector-Based Classification of Musical Phrases

Like in the case of investigating the musical phrase classification on the basis of the musical event sequence, the classification on the basis of mu-

sical phrase parameters by a classifier and a neural comparator was examined. The investigation concerning the neural classifier dealt with the individual musical phrase parametrization and the selected parameter combinations.

In the first stage the classification of training only with reference musical forms was performed. Table 4.23 presents data on the employed musical phrase parametrization and on the structures of employed neural networks as well as on the obtained classification results.

Table 4.23. Investigating parameter-based classification in the case of training with reference forms only

Parametrization method	Number of parameters	Number of neurons in the hidden layer	Number of iteration			
			1000		2000	
			MSE	Classification effectiveness - [%]	MSE	Classification effectiveness - average [%]
stat	5	20	0.94633	6.89	0.92082	12.34
stat	5	50	0.90056	14.26	0.90214	13.62
trig	5	20	0.10128	78.52	0.09826	78.53
trig	5	50	0.08625	85.26	0.06021	87.02
trig	10	20	0.10887	79.97	0.06554	83.81
trig	10	50	0.12693	80.61	0.10465	82.05
trig	15	20	0.04078	87.02	0.03516	87.02
trig	15	50	0.14704	78.85	0.10481	81.89
poly	5	20	0.20356	60.9	0.12456	68.75
poly	5	50	0.13287	68.59	0.08023	69.71
poly	10	20	0.02090	67.63	0.00542	66.19
poly	10	50	0.00987	64.90	0.00297	64.74
poly	15	20	0.11423	58.17	0.03661	60.26
poly	15	50	0.03961	62.18	0.00187	64.74
stat+trig	5+10	20	0.11657	80.77	0.09324	82.85
stat+trig	5+10	50	0.06371	87.82	0.04290	89.42
stat+poly	5+10	20	0.74441	23.24	0.73575	24.52
stat+poly	5+10	50	0.64372	29.17	0.68451	26.44
trig+poly	10+10	20	0.09703	81.25	0.07381	83.17
trig+poly	10+10	50	0.08508	84.62	0.04229	87.82
stat+trig+poly	5+10+10	20	0.07649	82.69	0.05310	84.29
stat+trig+poly	5+10+10	50	0.18616	76.44	0.10475	82.21

Leave-one-out tests were performed for a trigonometric parametrization. The process of network training was limited to 1000 iterations. Table 4.24 presents data on structures of the employed neural networks as well as on the obtained classification results for *leave-one-out* tests.

Table 4.24. Parameters of neural networks and classification results for *leave-one-out* tests on the basis of parametric representation of musical phrases. Best results for individual musical phrase modifications are marked as bold

			Musical phrase modification/ Classification effectiveness [%]									
Parametrization method	No. of parameters	No. of neurons in HL	No modif.	Error	Error+ transp.+ ornament	Error + transp.	Addit. note (beginning of the phrase)	Omission of the first note + ornament	Ornament	Transp.	Augm.	Augm. + transp.
trig	5	20	89.6	58.3	79.2	75	43.8	45.8	83.3	89.6	68.8	87.5
trig	5	50	100	87.5	85.4	93.8	50	62.5	97.9	100	91.7	95.8
trig	10	20	97.9	83.3	91.7	85.4	62.5	62.5	100	100	93.8	95.8
trig	10	50	97.9	89.6	95.8	91.7	70.8	68.8	93.8	100	95.8	93.8
trig	15	20	100	79.2	97.9	91.7	62.5	70.8	100	100	95.8	100
trig	15	50	97.9	91.7	93.8	100	58.3	58.3	95.8	100	91.7	97.9
trig	20	20	100	89.6	95.8	91.7	54.2	62.5	100	100	95.8	100
trig	20	50	100	91.7	93.8	100	79.2	68.8	100	100	95.8	100

4.2.8 Rough Set-Based Approach

The investigations employing the rough set approach were performed in the analogous way to the investigations concerning artificial neural networks. Experiments were performed for the TOFFEE database only. As said before, the investigations employed the ROSETTA system. On the basis of pilot runs, system operation parameters were limited to the following values:

- EFIM (*Equal Frequency Interval Method*) quantization, 10 intervals,
- generation of reducts and rules by means of a genetic algorithm.

Likewise in the case of investigations concerning artificial neural networks, intervals were represented using the binary method as well as a modified Hörnel's representation. Tests were run for collections both without representation of rhythmical data and with rhythm encoded using the binary representation. Values were buffered using the fading memory model.

Table 4.25 summarizes classification results obtained in the first stage of the presented investigation. Similar to investigations concerning artificial neural networks, *leave-one-out* tests were performed for a modified Hörnel's representation by means of rough sets. The effectiveness of clas-

sification as a function of rhythmic data representation was analyzed. The obtained classification results are presented in Table 4.26.

Table 4.25. Investigating classification using the rough set approach in the case of training with reference forms only

No rhythm representation				Rhythm representation			
No.	Database indicator	Fading coefficient	Classification effectiveness [%]	No.	Database indicator	Fading coefficient	Classification effectiveness [%]
1.	mbin_rn	0.8	85.6	7.	mbin_rrel	0.8	89.4
2.	mbin_rn	0.5	86.2	8.	mbin_rrel	0.5	88.3
3.	mbin_rn	0.2	82.2	9.	mbin_rrel	0.2	83.8
4.	mhor_rn	0.8	90.1	10.	mhor_rrel	0.8	92.6
5.	mhor_rn	0.5	87.8	11.	mhor_rrel	0.5	89.4
6.	mhor_rn	0.2	80.6	12.	mhor_rrel	0.2	81.6

Table 4.26. Investigating classification using the rough set approach in the case of training with reference forms only

Database indicator	Fading coeff.	Musical phrase modification/Classification effectiveness [%]									
		no modification	error	error transposition ornament	error transposition	additional note (beginning of the phrase)	omission of the first note ornament	ornament	transposition	augmentation	augmentation transposition
mhor_rn	0.8	100	81.3	79.2	83.3	95.8	83.3	70.8	100	100	100
mhor_rn	0.5	95.8	87.5	79.2	83.3	95.8	83.3	68.8	95.8	95.8	95.8
mhor_rn	0.2	91.7	81.3	92.9	68.8	91.7	70.8	64.6	91.7	91.7	91.7
mhor_rrel	0.8	100	89.6	81.3	89.6	97.9	85.4	66.7	100	100	100
mhor_rrel	0.5	100	95.7	95.4	89.6	97.9	85.4	66.7	100	100	100
mhor_rrel	0.2	91.7	91.7	79.2	81.3	87.5	85.4	72.9	91.7	91.7	91.7

Classification on the Basis of Parameters of Musical Phrases

Similar to investigations concerning artificial neural networks, a number of *leave-one-out* tests were performed for musical phrases represented by means of trigonometric parameters and RS method. The obtained results are presented in Table 4.27.

Table 4.27. Investigating classification using the rough set approach in the case of training with reference forms only

Parametrization method	Number of parameters	Musical phrase modification/Classification effectiveness [%]									
		no modification	error	error transposition ornament	error transposition	additional note (beginning of the phrase)	omission of the first note ornament	ornament	transposition	augmentation	augmentation transposition
trig	5	97.9	70.8	77.1	81.3	39.6	43.8	97.9	97.9	95.8	97.9
trig	10	100	87.5	85.4	81.3	43.8	52.1	100	100	97.9	97.9
trig	15	100	89.6	87.5	93.8	54.2	62.5	100	100	95.8	97.9
trig	20	100	89.6	91.7	91.7	56.3	64.6	100	100	100	100

Using the neural comparator to determine phrase identity, also the classification of musical phrases was investigated, employing trigonometric parameters. *Leave-one-out* tests were performed for phrase representations employing 5 and 10 trigonometric parameters. Data set was limited to six fugue themes randomly chosen from the collection. The obtained results are presented in Table 4.28.

Table 4.28. Results of *leave-one-out* tests for musical phrase comparator - rough set approach

Modification	Classification effectiveness [%]					
	5 parameters			10 parameters		
	negative	positive	average	negative	positive	average
error	100	100	100	100	83.3	98.5
error + transposition + ornament	100	83.3	98.5	100	100	100
error + transposition	100	100	100	100	100	100
additional note (phrase beginning)	100	66.7	97	100	83.3	98.5
omission of the first note + ornament	100	100	100	100	50	95.5
ornament	100	66.7	97	100	100	100
transposition	100	83.3	97.6	100	83.3	97.6
augmentation	100	100	100	100	100	100
augmentation + transposition	100	100	100	100	100	100

Discussion of Results

A number of conclusions concerning the developed methods of representing and automatically classifying musical phrases can be drawn on the basis of the experiments performed. The methods developed enable the effective classification of musical phrases in the presence of phrase modifications which are characteristic for the techniques of composing and performing. Artificial neural networks and the rough set-based approach show comparable suitability for classifying musical phrases on the basis of sequence data and of musical phrase parameters. Remarks concerning the musical data representation are as follows, for the classification on the basis of sequence data, the highest classification effectiveness was obtained for the modified Hörnel's representation and a relative rhythm representation. On the other hand for feature vectors containing parameters, the best results were obtained for trigonometric parametrization. The obtained results indicate also changes of data entropy depending on a musical form and a composer's style, this may result in the possibility of predicting musical data with high accuracy in the case of exact polyphonic forms.

High classification correctness obtained with intelligent decision algorithms and musical sequence representations using the fading memory model was negatively verified in the conditions of musical material homogeneity. The decision process employing rough sets revealed the best ratio of classification accuracy to computational complexity.

4.2.9 Automatic Retrieval of Rhythmic Patterns

Yet, another investigation was carried out to check whether it is possible to automatically retrieve rhythmic patterns in a melody line. This research is the subject of the Ph.D. thesis of Wojcik, the author's Ph.D. student (Kostek and Wojcik 2004; Wojcik and Kostek 2004; Wojcik et al 2004).

A lot of research was done in the area of melody retrieval (http://www.ismir.net; Tseng 1998, 1999; Wu and Manber 1992). Melody retrieval systems, as presented before, can now accept hummed queries and retrieve melodies even though users make musical mistakes in queries. Contrarily to melody-based information retrieval, the area of music information retrieval concerning rhythm has not been well explored. Scientists search for a characteristic rhythmic pattern of the known length in a piece (Chin and Wu 1992) or try to find length and onsets of rhythmic patterns for a given melody (Dixon 2001; Rosenthal 1992a, 1992b). Other works on rhythm finding are, among others, by Povel and Essens (1985) or Parncutt (Parncutt 1994). Most of approaches are based on a generative theory of tonal music (Lerdahl and Jackendoff 1983). Dixon's (2001) and

Rosenthal's (1992a, 1992b) systems form and then rank the rhythmical hypotheses, taking into account mainly musical salience of sounds. However salience functions proposed by Dixon and Rosenthal are based on human intuition only. It is also possible to adopt artificial intelligence learning techniques to find salience functions on the basis of real-life musical files. They are employed to estimate rhythmical salience of sounds in a melody, then the knowledge obtained may be used to improve approaches to the hypotheses of ranking, which would eventually result in finding a proper rhythm to a given melody.

Self-organizing networks can learn to detect regularities and correlation in their input and adapt their future responses to that input, accordingly. The neurons of competitive networks (layers) learn to recognize groups of similar input vectors. Learning Vector Quantization (LVQ) is a method for training competitive layers in a supervised manner. A competitive layer automatically learns to classify input vectors into subclasses, then a linear layer transforms them into target classes chosen by the user. The subclasses that are found by the competitive layer are dependent only on the distance between input vectors. Thus, if two input vectors are very similar, the competitive layer probably puts them into the same subclass.

The LVQ network consists of two layers: the first is a competitive layer, the second one is a linear layer. The competitive layer learns to classify input vectors into subclasses. In the beginning, the negative distances between the input vector and the input weight (IW) vectors are calculated. The distance vector consists of n_c elements, where n_c is the number of neurons in the competitive layer. The net input vector is the sum of the distance vector and the bias vector. Depending on the bias vector, the competitive transfer function finds an appropriate subclass by seeking out the most positive or the least negative value in the network input vector. If all elements of the bias vector are zeros, then the output subclass number is the position of the least negative value in the net input vector, otherwise the output subclass number is the position of the most positive value of that vector. The role of a bias is to balance the activation of neurons. This causes dense regions of the input space to be classified as more subsections. Both the competitive and linear layers have one neuron per subclass or per target class. Thus, the competitive layer can learn up to n_c subclasses. These, in turn, are combined by the linear layer to form n_t target classes. The value of n_c is always greater than n_t. For example, let us take neurons 1, 2, and 3 of the competitive layer into consideration, they all learn subclasses of the input space that belongs to the linear layer target class No. 2. Then competitive neurons 1, 2 and 3 have weights of 1 to neuron n_2 from the linear layer, and weights of 0 to all other linear neurons. Thus, the linear neuron produces 1 if any of the three competitive neurons

(1, 2 and 3) wins the competition. This is the way the subclasses of the competitive layer are combined into target classes in the linear layer.

LVQ networks are trained in a supervised manner. Therefore learning data consist of pairs $\{p,t\}$ where p and t are input and desired output vectors respectively. The output vector t consists of values 0 and a single value of 1 placed at the position corresponding to the number of the class a given element p belongs to. During the qth epoch vector $p(q)$ is presented at the input, and then the output from network a_2 is compared to t. Let i^* be a position in t where 1 occurs and j^* the position where 1 occurs in t. If $i^*=j^*$, then p is classified correctly, then:

$$_{i^*}IW^{1,1}(q)=_{i^*}IW^{1,1}(q-1)+\alpha(p(q)-_{i^*}IW^{1,1}(q-1)) \tag{4.68}$$

otherwise (p classified incorrectly)

$$_{i^*}IW^{1,1}(q)=_{i^*}IW^{1,1}(q-1)-\alpha(p(q)-_{i^*}IW^{1,1}(q-1)) \tag{4.69}$$

The i^*th row of the IW matrix is adjusted in such a way as to move this row closer to the input vector p if the assignment is correct, and to move it away from p otherwise. Described corrections made to the i^*th row of $IW^{1,1}$ can be made automatically without affecting other rows of $IW^{1,1}$ by backpropagating the output errors to layer 1. Such corrections move the hidden neuron towards vectors that fall into the class for which it forms a subclass, and away from vectors that fall into other classes.

It is possible to estimate musical salience taking into account the physical attributes of sounds in a melody, based on the Data Mining association rule model, proposed by (Mannila 1996). This approach explores a training data set and finds tendencies, which determine the knowledge used to predict most probable associations between attributes in a testing set. If the tendencies discovered are confirmed in tests, the knowledge obtained can be used for other melodies to rank rhythmical hypotheses.

In the association rule model there is a set of attributes in learning table T, some of which are classifying attributes. The rows of the table are training objects. In this approach the row is a sound. Attributes can have Boolean values 0 or 1. A rule is a statement saying that the presence of values of 1 in a certain set of attributes usually causes the classifying attribute to have the value of 1 as well. An example of a rule can be "long sounds tend to be placed in accented positions of the musical piece". A statement X → Y can be acknowledged as a rule if its confidence in table T has higher values than other rules.

$$confidence\ (X\rightarrow Y,T) = support\ (X\rightarrow Y,T)\ /frequency(X,T) \tag{4.70}$$

where:

$$support(X \rightarrow Y,T) = frequency\ (X \cup Y,T) \tag{4.71}$$

and

$$frequency\ (Z,T) = z / t \tag{4.72}$$

where *z* is a number of records in table T, whose all attributes from set Z have the value of 1. Set X is a so-called premise of a rule, and set Y is a conclusion of this rule.

While employing Artificial Neural Networks, the learning and testing sets were created from MIDI files of various styles. Data were divided in two databases:

- single-track melodies: 20 351 sounds divided into 10 learning and 10 testing sets,
- multi-track musical pieces: 42 998 sounds divided into 21 learning and 21 testing sets.

The size of testing sets was averagely 2.5 times larger than the learning ones. Music files were obtained from the Internet using a web robot. For the purpose of training accented locations in each melody have been found. During subjective listening tests, musical pieces with wrongly marked accented locations from the learning set have been removed. Also, non-melody tracks consisting of sounds from rhythmic instruments such as drums and bass guitars have been rejected from both learning and testing sets. One of tested networks had three inputs – one for each physical attribute of a sound (duration, frequency and amplitude), the second group consisted of three networks having one input for each physical attribute of a sound. Each attribute had a value from the range of 0 to 127. A desired output of the network could adopt one of two values – 1, if the sound was accented, or 0 if it was not.

In the testing stage the LVQ network determines whether a sound is accented or not according to the knowledge received in the learning stage. Outputs given by such a network are compared to real outputs. After the testing phase, the set of all sounds can be divided into four subsets (see Table 4.29), because there are four possible combinations of the desired outputs and network outputs.

Table 4.29. Possible combinations of real and network outputs

Description	Desired output	Network output
1. Sound not accented, accurately detected by a network	0	0
2. Sound not accented but falsely detected as accented	0	1
3. Sound accented, not detected	1	0
4. Sound accented, accurately detected	1	1

The network accuracy is formulated as a ratio of the number of accented sounds, which were accurately detected by the network (the number of elements in subset 4), to the number of accented sounds in a melody (the number of elements in the sum of subsets 3 and 4).

$$\textit{network accuracy} = \textit{number of accurately detected accented sounds/number of all accented sounds} \tag{4.73}$$

Since the network accuracy depends on the number of accents given by the network, a so-called hazard accuracy was introduced, which determines how accurately accented sounds could be found, if they were hit in a randomized way. The number of times the network recognizes accented sounds better than a 'blind choice' becomes clear after dividing the network accuracy by the hazard accuracy. This approach also helps to determine how well the network recognizes accented sounds if it takes into consideration single physical attributes or three of them simultaneously. The hazard accuracy depends on the number of accents given by the network (the number of elements in the sum of subsets 2 and 4) and the number of all sounds in a set (the number of elements in the sum of all four subsets).

$$\textit{hazard accuracy} = \textit{number of accented sounds detected by the network/ number of all sounds} \tag{4.74}$$

Single-track melodies were used as a learning/testing set. Fig. 4.21 presents how accurately four networks found the accented sounds. There are three lines presenting the results of networks fed only with one attribute and one line representing the network fed with all three physical attributes. The network accuracy was tested also on single-track melodies.

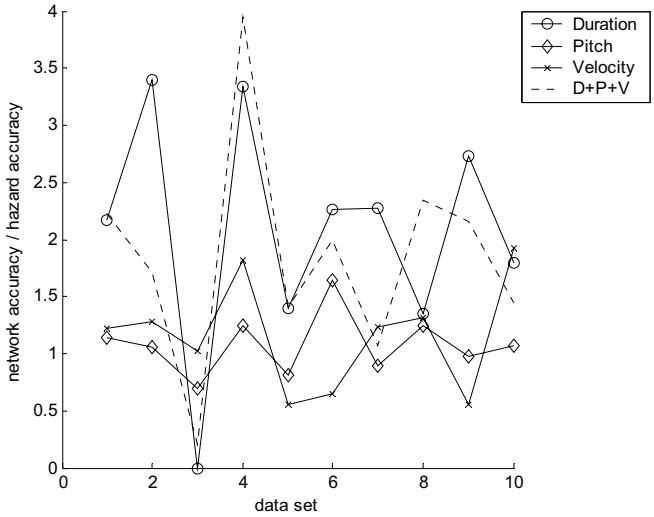


Fig. 4.21. Accuracy of four networks for single-track melodies (10 training sets and 10 testing sets)

The analogical experiment was conducted on multi-track musical pieces. Results of this experiment are shown in Fig. 4.22.

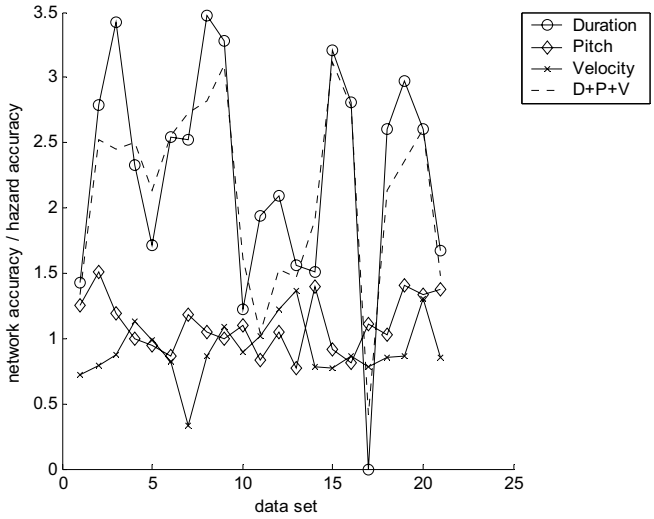


Fig. 4.22. Accuracy of four networks for multi-track musical pieces (21 training sets and 21 testing sets)

The above results have been averaged in order to get comparable, single numbers assigned to each of the networks. Standard deviation values (see Table 4.30) have been calculated and average values have been divided by standard deviations. Such a fraction helps to compare the stability of the results obtained for all networks. The lower the value of the fraction, the more stable the results. The results of this experiment will be discussed further in this study.

Table 4.30. Accuracy of networks tested on single-track melodies – mean values, standard deviations of network accuracy / hazard accuracy

		Network 1 Duration	Network 2 Frequency	Network 3 Amplitude	Network 4 Dur.+Frq. +Ampl.
10 sets	Mean (average) value	2.07	1.08	1.16	1.85
	Standard deviation	1.01	0.27	0.48	0.98
	Standard deviation/ Mean value	0.49	0.24	0.42	0.53
21 sets	Mean (average) value	2.27	1.10	0.91	2.12
	Standard deviation	0.86	0.21	0.22	0.72
	Standard deviation/ Mean value	0.38	0.20	0.24	0.34

As seen from these results, duration is the only attribute, which should be considered in the ranking hypothesis phase. The neural network accuracy based on frequency and amplitude oscillates around the hazard accuracy, those two attributes are too much dependent on the choice of a learning/testing data set. Results of experiments for single track melodies and multi-track musical pieces are consistent. In real music retrieval systems, it seems to be reasonable to take only sound duration into account – either for melodies or polyphonic, multi-instrumental pieces. Rhythmic salience depends on physical attributes in a rather simple way. Neural Networks fed with a combination of attributes performed even slightly worse than the ones fed with a single attribute (duration).

The Data Mining association rule model was also used to estimate salience of sounds in a melody. The testing set contained 36 966 sounds from 191 melodies. Since in the association rule model, attributes contained in a learning table can adopt only 1 or 0 values, thus they had to be preprocessed. Discretization was performed for these three attribute values. The equal subrange quantization method was employed. For each physical attribute its minimum and maximum values in a piece have been found and

subranges have been divided by thresholds placed according to the formula:

$$MinValue + (MaxValue - MinValue) \cdot j/m \text{ for } j = 0, 1, 2 \ldots m, \quad (4.75)$$

where m-1 is the number of subranges.

Table 4.31 presents a short example of preprocessed data for three subranges.

Table 4.31. Example of preprocessed data (3 subranges)

	Duration			Amplitude			Frequency			
sound	short	med.	long	quiet	med.	loud	low	med.	high	Accented?
	0	1	2	3	4	5	6	7	8	
1		1		1				1		
2			1		1				1	1
3	1				1			1		

The experiments have been performed for 10 various numbers of subranges of the range from 3 to 100. Using the formulae introduced earlier, 10 tables with information about all rules have been received, along with their supports and confidences in table columns. Rows of these tables can be treated as rules, the number of rows of each table is $3m$+1. It is possible to find a rule or rules with the highest confidence for each physical attribute. An example of rules of maximum confidences for each physical attribute is shown in Table 4.32. For example a rule 'high → accented' in Table 4.32 means that sounds of the low and medium frequency do not appear in accented positions as often as sounds with high frequency.

Table 4.32. Example of maximum confidence rules (3 subranges)

Premise	Conclusion	Support	Confidence
duration – long	accented	0.099	0.714 = $MaxC_{\text{dur}}$
frequency – high	accented	0.083	0.605 = $MaxC_{\text{frq}}$
amplitude – loud	accented	0.061	0.442 = $MaxC_{\text{amp}}$

These rules along with their confidences have been used to propose ranking hypothesis functions. All maximum confidences from Table 4.32 have been summed up. The received sum is called SMC – sum of maximum confidences. The value of each physical attribute of a sound, for which a salience function value is calculated, is first quantized using the same number of subranges as in the learning stage. Let the subranges where the value falls off be i_{dur}, i_{frq} and i_{amp}. After reading the values of

confidences C from the table of all rules, $C(i_{dur})$, $C(i_{frq})$ and $C(i_{amp})$ are known. The following ranking functions basing on the Data Mining knowledge may be proposed:

$$RANK_1 = [C(i_{dur}) + C(i_{frq}) + C(i_{amp})]/SMC$$

$$RANK_2 = [C(i_{dur})/MaxC_{dur} + C(i_{frq})/MaxC_{frq} + C(i_{amp})/MaxC_{amp}]/m$$

$$RANK_3 = C(i_{dur})/MaxC_{dur} \tag{4.76}$$

$$RANK_4 = C(i_{frq})/MaxC_{frq}$$

$$RANK_5 = C(i_{amp})/MaxC_{amp}$$

The first two functions take into account all physical attributes simultaneously, the remaining ones are consistent with $RANK_1$ and $RANK_2$, but they consider attributes separately. The values of all above formulae are normalized, they fall within the interval <0,1>.

These formulae have been compared with the ones proposed in the related research. In the Dixon's approach (Dixon 2001), two combinations of physical attribute values of sounds are proposed – a linear combination (additive function) and the multiplicative function.

$$s_{add}(d, p, v) = c_1 \cdot d + c_2 \cdot p[p_{\min}, p_{\max}] + c_3 \cdot v \tag{4.77a}$$

$$s_{mul}(d, p, v) = d \cdot (c_4 - p[p_{\min}, p_{\max}]) \cdot \log(v) \tag{4.77b}$$

where:

$$p[p_{\min}, p_{\max}] = \begin{cases} p_{\min}, & p \le p_{\min} \\ p, & p_{\min} < p < p_{\max} \\ p_{\max} & p_{\max} \le p \end{cases} \tag{4.78}$$

In the Dixon's formulae p_{min}, p_{max} and c_k (k=1,2,3,4) are experimentally adopted constants, d is a duration of a sound, p is MIDI pitch (a frequency) and v is a MIDI velocity of a sound (an amplitude).

The precision/recall method to validate the accuracy of each of the above given functions has been used.

$$Precision = number\ of\ relevant\ documents\ in\ answer/number\ of\ documents\ in\ answer \tag{4.79}$$

Recall = number of relevant documents in answer / number of relevant documents in database (4.80)

In such an evaluation a single sound plays a role of a document, and an accented sound is a relevant document. Sounds are sorted descending according to the value of each ranking function. The highly ranked sounds are placed in the answer. The number of sounds placed in the answer equals the number of the relevant documents (sounds placed in accented positions). This results in the equality of precision and recall giving a single measure, making the comparison of ranking approaches easy. Fig. 4.23 presents the accuracy of all seven described ranking functions.

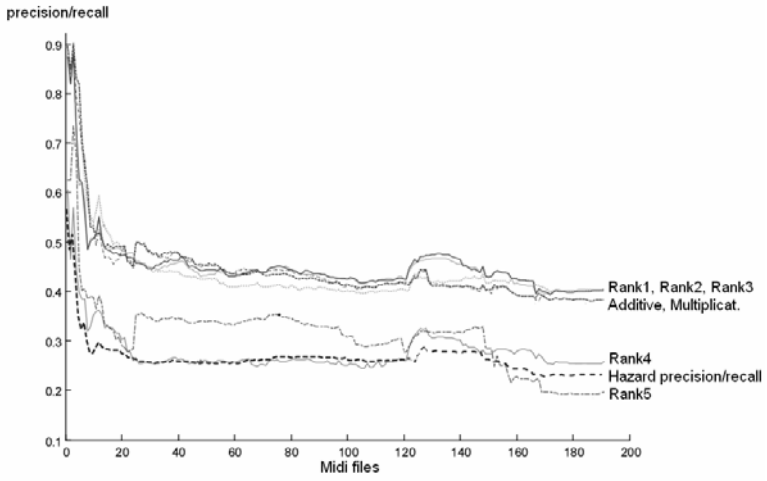


Fig. 4.23. Precision/recall of retrieval – ranking functions approach

In these experiments precision/recall values have been first counted for one piece, then sounds from one more piece have been added to the learning table and precision/recall values have been counted again. This action was repeated for all the pieces used in experiments. This was to check, whether the value of precision/recall stabilizes.

On the basis of Fig. 4.23, it may be concluded that ranking functions awarding duration retrieve accented sounds much better than $RANK_4$ and $RANK_5$ functions, which take into consideration frequency and amplitude, respectively. Also, long sounds tend to be accented – Table 4.32 presents the rules of maximum confidences. The precision/recall of $RANK_3$ stabilizes after adding about 30 pieces to the testing/training Data Mining set, thus the duration-based salience of a sound can be considered as certain. $RANK_4$ and $RANK_5$ loose stability even after the 120th piece – those two

attributes are very dependent on training/testing data. Consequently, $RANK_1$, $RANK_2$ and both Dixon's precision/recall formulae are less stable in those locations. This is why using duration can be recommended only in calculating the sound rhythmic salience.

In the experiment of the association rule, performance of salience functions appeared to depend on the number of discretization subranges. In experiments with larger numbers of subranges the system performance grew up from 3 up to about 30 subranges. For learning/testing datasets preprocessed with more than 30 subranges further growth of performance was not observed. Relations between performances of functions awarding duration, frequency and amplitude remained unchanged, however.

Another approach to rhythmic value retrieval included a rough set-based method. It occurs that the RSES system (Bazan and Szczuka 2001; Bazan et al 2002) used in experiments returns quite precisely the information about sounds that are not accented, on the other hand, the decision about accented sounds is correct only in 50% of cases. This signifies that additional temporal descriptors should be included in feature vectors for making an adequate decision. In Tables 4.33 and 4.34 the results of the RSES system performance are shown. Table 4.33 presents results obtained with the global discretization employed, and Table 4.34 shows results for the local discretization. As observed from both tables, if the actual sound was not accented, the RSES system recognizes this fact with quite a high accuracy (corresponding values are 0.815 and 0.85). On the other hand, a decision pointing at the accented sounds is at the level of 50%, as mentioned previously, thus it is more like a hazardous guess.

Table 4.33. Testing rhythmical datasets with the RSES system (global discretization)

Global discretization					
	Predicted				
Actual	1	0	No. of obj.	Accuracy	Coverage
1	**1391**	1235	2626	0.530	0.799
0	1690	**7446**	9136	0.815	0.845

Table 4.34. Testing rhythmical datasets with the RSES system (local discretization)

Local discretization					
	Predicted				
Actual	1	0	No. of obj.	Accuracy	Coverage
1	**1414**	1177	2591	0.545	0.789
0	1233	**6993**	8226	0.850	0.760

Further steps in the research include more temporal descriptors in the datasets, along with the information about periodicity, and testing of these data with the above presented systems. This especially concerns testing the quality of feature selection, the optimum choice of the number of subranges and the quality of rules derived.

4.3 MUSIC ANALYSIS

A broadened interest in music information retrieval from music databases, which are most often heterogeneous and distributed information sources, is based on the fact that they provide, apart from music, a machine-processable semantic description. The semantic description is becoming a basis of the next web generation, i.e., the Semantic Web. Several important concepts were introduced recently by the researchers associated with the rough set community with regard to semantic data processing including techniques for computing with words (Komorowski et al 1998; Pal et al 2004). Moreover, Zdzislaw Pawlak in his recent papers (Pawlak 2003, 2004) promotes his new mathematical model of flow networks which can be applied to mining knowledge in databases. Given the increasing amount of music information available online, the aim is to enable effective and efficient access to such information sources. A concept was introduced that covers the following issues: the organization of a database, the choice of a searching engine, and the detail information on how to apply the whole conceptual framework based on flow graphs in order to achieve better efficiency in the retrieval of music information.

The experiments that were performed consisted in constructing a music database that collects music recordings along with semantic description. A searching engine is designed, which enables searching for a particular musical piece. The knowledge on the entire database content and the relations among its elements contained in the flow graphs constructed following Pawlak's ideas are utilized in this search process.

Generally, the study addresses the capabilities that should be expected from intelligent Web search tools in order to respond properly to the user's needs of multimedia information retrieval. Two features, seem to be of great importance for searching engines: the ability to properly order the retrieved documents and the capability to draw the user's attention to other interesting documents (intelligent navigation concept). These goals could be efficiently achieved provided the searching engine uses the knowledge of database content acquired a priori and represented by distribution ratios

between branches of the flow graph which in turn can be treated as a prototype of a rule-based decision algorithm.

4.3.1 Database Organization

Data Description in CDs

One of the most important music archiving format is the data format of CDs (Compact Disks). According to so-called Red Book specifications (ICE 908), a CD is divided into a lead-in area, which contains the table of contents (TOC), a program area, which contains the audio data, and a lead-out area, which contains no data. An audio CD can hold up to 74 minutes of recorded sound, and up to 99 separate tracks. Data on a CD is organized into sectors (the smallest possible separately addressable block) of information. The audio information is stored in frames of 1/75 second length. 44.100 16-bit samples per second are stored, and there are two channels (left and right). This gives a sector size of 2,352 bytes per frame, which is the total size of a physical block on a CD. Moreover, CD data is not arranged in distinct physical units; data is organized into frames (consisting of 24 bytes of user data, plus synchronization, error correction, and control and display bits) which are interleaved (Pohlman 1992).

CDDB Service

CDDB service is the industry standard for music recognition services. It contains the largest online database of music information in the world (currently more than 22 million tracks), and is used by over 30 million people in more than 130 countries every month. Seamless handling of soundtrack data provides music listeners, both professional and amateurs, with access to a huge store of information on recorded music (http://www.freedb.org; http://www.gracenote.com). The large database queried so frequently by users from all over the world provides also a very interesting material for research experiments in the domain of the optimization of searching engines. The organization of metadata related to compact discs in the CDDB database is presented in Table 4.35.

The content of the world-wide CDDB was targeted in the experiments as the principal material for experiments. However, because of the large volume of this database and the expected high computational cost, it was decided that a much smaller local database utilizing the CDDB data format would be constructed at the initial stage. Consequently, a database was constructed especially for the purpose of this study containing approxi-

mately 500 compact discs textual data stored together with fragments of music corresponding to various categories. This database provided a material for initial experiments on searching music employing the proposed method. Subsequently, the huge CDDB database containing metadata related to majority of compact disks hitherto produced in the world was utilized.

Table 4.35. Metadata fields in the CDDB database

Album Data Fields:	
Album Title	Can be a multi-word expression (string)
Album Artist	as above
Record Label	The label or publisher of the CD
Year	The year the CD was recorded or published
Genre	Every album can have both a primary and a secondary genre
Compilation	Indicates whether tracks have different artists
Number/Total Set	Can identify a CD as a member of a box sets
Language	Used to help display in appropriate character set
Region	To identify where the CD was released
Certifier	Authorized party (artist or label) who has certified the data accuracy
Notes	General notes such as dedications, etc.

Track Data Fields:	
Track Title	Can be a multi-word expression (string)
Track Artist	Vital for compilations, such as soundtracks or samplers
Record Label	May be different from track to track for compilations
Year	May be different from track to track
Beats/Minute	Used for special purposes (synchronizing with special devices)
Credits	E.g. guest musicians, songwriter, etc.
Genre	Every track can have both a primary and a secondary genre
ISRC	The International Standard Recording Code number for the CD track
Notes	General track notes such as “recorded in ...”, etc.
Credits	Can be entered for entire album, for individual tracks or segments
Credit Name	Can be person, company, or place such as recording location
Credit Role	Instrument, composer, songwriter, producer, recording place, etc.
Credit Notes	E.g. to specify unusual instruments, etc.
Genres	Can be entered for entire album or applied to individual tracks
Metagenres	General classification. e.g. Rock; Classical; New Age; Jazz
Subgenres	More specific style. e.g. Ska; Baroque, Choral; Ambient; Bebop, Ragtime
Segments	Each segment can have its own name, notes, and credits

Extended Tagging System

The information presented in this paragraph is not directly related to the current experimental phase. Nevertheless, it provides an extension to current standards, illustrating the future trends and expected developments of methods for advanced music searching in databases that use metadata associated with musical recordings.

The ID3v2 is a currently used tagging system that enables to enrich and to include extended information about audio files within them (Fig. 4.24). It represents data prepended to binary audio data. Each ID3v2 tag holds one or more frames. These frames can contain any kind of information and data, such as a title, an album, a performer, Website, lyrics, equalizer pre-sets, pictures etc. Since each frame can be 16MB and the entire tag can be 256MB thus there is a lot of place to write a useful comment. The ID3v2 supports Unicode, so that the comments can be written in the user's native language. Also information on the language of comments can be included (http://www.id3.org). In addition the main characteristics of this system are as follows:

- It is a container format allowing new frames to be included.
- It has several new text fields such as a composer, a conductor, a media type, a copyright message, etc. and the possibility to design user's own fields.
- It can contain lyrics as well as music-synced lyrics (karaoke) in many languages.
- It could be linked to CD-databases such as CDDB (http://www.freedb.org)
- It supports enciphered information, linked information and weblinks.

An example of the internal layout of an ID3v2 tagged file is presented in Fig. 4.24.

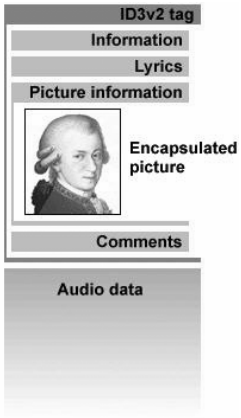


Fig. 4.24. Example of the internal layout of an ID3v2 tagged file (http://www.id3.org)

In comparison to the CDDB format, a much larger and more diversified metadata set of ID3v2 standard opens a way towards future experiments in

the domain of advanced music searching including application of the method proposed in this paper.

4.3.2 CDDB Database Organization and Searching Tools

A sample record from the CDDB database is presented in Fig. 4.25. The field denoted as “frames” needs some explanation. It contains the frame numbers, because the CDDB protocol defines the beginning of each track in terms of track lengths and the number of preceding tracks. The most basic information required to calculate these values is the CD table of contents (the CD track offsets, in "MSF" [Minutes, Seconds, Frames]). That is why tracks are often addressed on audio CDs using "MSF" offsets. The combination determines the exact disc frame where a song starts.

Tools for CDDB Information Retrieval

The process of querying the CDDB database begins when the unique content of the “frames” field is issued to the database searching engine. In response, the CDDB database transmits back all the data related to the album – namely its artist, title,..., genre, etc. This feature is exploited by a huge number of users world-wide. However, such a query can be executed provided that users possess a copy of the CD record whose metadata are searched for. If so, their computers can automatically get the data from the CDDB database and display information as illustrated in Fig. 4.26. Consequently, local catalogs of records can be built up fast and very efficiently with the use of this system.

A possible benefit from the universal and unrestricted access to CDDB could be, however, much greater than just obtaining the textual information while having a copy of a record at a disposal. Namely, provided an adequate searching engine is employed, CDDB users could submit various kinds of queries to this largest set of data on recorded sound, without the necessity to gain an access to any CD record in advance. The purpose of such a data search mode could be different and much broader than building up catalogs of available records – it could have various research, historic and cultural applications and connotations. The currently available searching engines are able to scan CDDB content for keywords or keyword strings. Usually, if the query sentence consists of several words, the logical operator AND is utilized (Boolean searching model). This, in many occasions, brings very poor results, because too many matches occur in the case of famous or there are no matches at all if the operator enters terms that are misspelled or mixed-up. An important feature of the searching en-

gine is the ability to order matches according to users' expectations and to adapt properly if attributes are close, but not necessarily exact. The latter assumption is easy to illustrate on a typical example presented in Fig. 4.27. The example shows that the CDDB database contains many records related to the same CD. That is because all CDDB users possessing records are allowed to send and store remotely metadata utilizing various software tools (as shown in Fig. 4.28). Consequently, textual information related to the same CD records can be spelled quite much differently.

CDDBID: eb117b10
[22164FD]
artist=Céline DION
title=Let's Talk About Love
numtracks=16
compilationdisc=no
genre=Pop
year=1997
comment=
0=The Reason
1=Immortality
2=Treat Her Like A Lady
3=Why, Oh Why ?
4=Love Is On The Way
5=Tell Him (Avec Barbra Streisand)
6=Amar Haciendo El Amor
7=When I Need You
8=Miles To Go (Before I Sleep)
9=Us
10=Just A Little Bit Of Love
11=My Heart Will Go On (Chanson D'amour Du Film Titanic)
12=Where Is The Love ?
13=Be The Man
14=I Hate You Then I Love You (Avec Luciano Pavarotti)
15=Let's Talk About Love
frames=0,22580,41415,59812,81662,101655,123540,142347,161295,182290,208287,226792,
247817, 270010,290987,312245,335675
order=0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Fig. 4.25. Sample CDDB database record

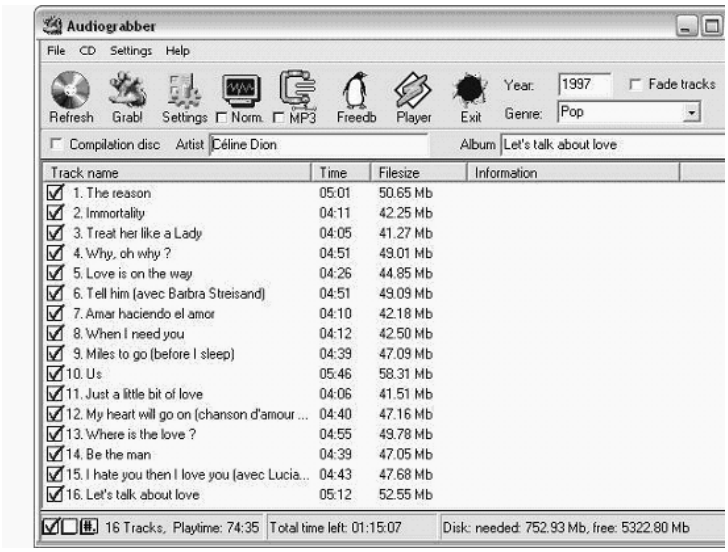


Fig. 4.26. Screen shot from the Audiograbber software (http://www.audiograbber.com-us.net/) used for CDDB querying and fast audio track copying. The presented data set corresponds to the database record as in Fig. 4.25

Artist	Album
Celine Dion	Let's Talk About Love
Céline Dion	Let's talk about love
Celine Dion	Let's Talk About Love
CELINE DION	CELINE DION LET'S TALK ABOU...

Fig. 4.27. Sample result of information retrieval from the CDDB database obtained after sending content of "frames" field (4 different records were retrieved concerning the same disk)

Fig. 4.28. Panel of typical software tool allowing users to transmit & store metadata in the CDDB database

Simple Search Questionnaires

An important factor in building a query questionnaire for a music database is the knowledge of how the users access music information. The question arises whether all data contained in the album and track data fields are used by a person searching for a particular recording or a CD. A survey on this problem was performed by Bainbridge et al. (2003). Some of these results are recalled in Table 4.36. It is most common that a search starts with the name of an album or a performer. Next in popularity are searches by title and date of recording. It has also been discovered that users are rarely able to fill in such a questionnaire with data, which are exact to ones stored in a music database. They experience difficulty in coming up with crisp descriptions for several of the categories. This indicates a need to support imprecise metadata values for searches. Apart from not being able to give crisply defined categories, users often make mistakes, or are not certain when referencing to the year of a recording. They would rather give the decade or define the time in other terms than a singular year. Also giving information on lyrics, may cause some problems, such as how to transcribe non existing words. Another category, which is difficult for users is the genre of recording. All these factors should be taken into account when building a questionnaire for a database search.

Meanwhile, despite a really expressive information that can be found on some Webpages (http://www.gracenote.com) available searches are very simplistic, basing just on text scanning with an application of simple logical operators in the case of multi-term queries (Boolean search). The panel of such an electronic questionnaire was reproduced in Fig. 4.29.

Table 4.36. Survey on user's queries on music (Bainbridge et al 2003)

Category	Description	Count	[%]
Performer	Performer or group who created a particular recording	240	58.8
Title	Name (or approximation) of work(s)	176	43.1
Date	Date that a recording was produced, or that a song was composed	160	39.2
Orchestration	Name of instrument(s) and/or vocal range(s) and/or genders (male/female)	68	16.7
Album title	Name of album	61	15.0
Composer	Name of composer	36	8.8
Label	Name of organization which produced recording(s)	27	6.6
Link	URL providing a link to further bibliographic data	12	2.9
Language	Language (other than English) for lyrics	10	2.5
Other	Data outside from the above categories	36	8.8

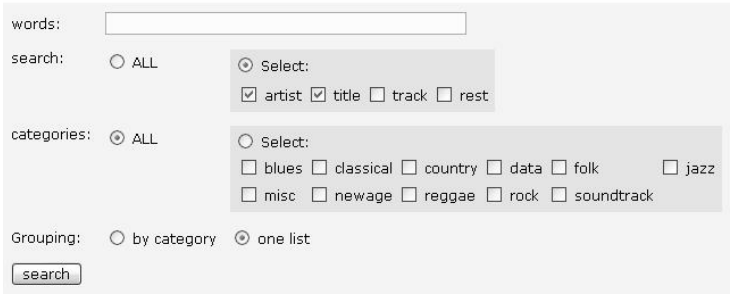


Fig. 4.29. Freedb.org simple searcher interface

Advanced Metadata Searching Attempts

Recently, a rapid growth of interest is observed in the so-called 'Semantic Web' concepts (http://www.semanticweb.org/). By the definition 'Semantic Web' the representation of data on the World Wide Web is understood. The effort behind the Semantic Web is to develop a vocabulary for a specific domain according to the recent ISO/IEC 13250 standard, which aims at capturing semantics by providing a terminology and link to resources. It is based on the Resource Description Framework (RDF), which integrates a variety of applications using XML for syntax (http://www.w3.org/2001/sw/). In other words, these are sets of organizing guidelines for publishing data in such a way that it can be easily processed by anyone. Within the work done on the Semantic Web automated reasoning, processing, and updating distributed content descriptions are also discussed. A so-called problem solving method (PSM) approach is taken in which algorithms from the field of Knowledge Based Systems are considered to perform inferences within expert systems. However, PSM are task specific. At the moment there seems to be little consensus on the characteristics of the Semantic Web, but many domains are already covered by some applications and test databases, or at least guidelines have been created.

Recent developments in the MPEG-7 and MPEG-21 standards enable to create multimedia content descriptions, which includes audio and video features at the lower abstraction level as well as at the higher conceptual level (http://www.darmstadt.gmd.de/mobile/MPEG7/; Lindsay and Herre 2001). Low-level features of the content have been already thoroughly reviewed in Chapter 3. On the other hand, higher level descriptors can be used to provide conceptual information of the real world being retrieved by the content. Intermediate levels of description can provide models that

link low-level features with semantic concepts. Because of the importance of the temporal nature of multimedia, dynamic aspects of content description need to be considered. Therefore, the semantic multimedia web should be dedicated to conceptual and dynamical aspects of content description.

Even though a lot of research was focused on low-level audio features providing query-by-humming or music retrieval systems (Downie 1999, 2003a, 2003b; Herrera et al 2000; http://www.ismir.net/; Kostek 1999, 2004a, 2004b; Kostek and Czyzewski 2001, Pfeiffer et al 1996), most web sites containing music support only category or text-based searching. Few reports can be found on attempts to build advanced tools for mining the content of the CDDB database, i.e. the study by Bainbridge (2003), Downie (2003b), and Pachet et al. (2002). One of the reports on this subject was published by Pachet et al. (2000). They propose a method of classification based on musical data mining techniques that uses a co-occurrence and correlation analysis for classification. In essence, the authors of the cited work concentrated on processing the extracted titles and on establishing similarity measurements between them. The co-occurrence techniques were applied to two data sources: radio program, and CD compilation database.

The co-occurrence analysis is a well-known technique used to statistical linguistics in order to extract clusters of semantically related words. In the case of the study by Pachet et al. the analysis consisted in building a matrix with all titles in rows and columns. The value at (*i, j*) corresponds to the number of times that titles *i* and *j* appeared together, either on the same sampler, or on the same web page, or as neighbors in a given radio program.

Given a corpus of titles $S = (T_1,...,T_N)$, the co-occurrence may be computed between all pairs of titles T_i and T_j, (Pachet et al 2002). The co-occurrence of T_i with itself is simply the number of occurrences of T_i in the considered corpus. Each title is thus represented as a vector, with the components of the vector being the co-occurrence counts with the other titles. To eliminate frequency effects of the titles, components of each vector are normalized according to:

$$Cooc_{norm}(T^1,T^2) = \left(\frac{Cooc(T^1,T^2)}{Cooc(T^1,T^1)} + \frac{Cooc(T^2,T^1)}{Cooc(T^2,T^2)} \right) / 2 \tag{4.81}$$

The normalized co-occurrence values can be directly used to define a distance between titles. The distance will be expressed as:

$$Dist_1(T^1,T^2) = 1 - Cooc_{norm}(T^1,T^2) \tag{4.82}$$

Given that the vectors are normalized, one can compute the correlation between two titles T^1 and T^2 as:

$$Sim(T^1, T^2) = \frac{Cov_{1,2}}{\sqrt{Cov_{1,1} \times Cov_{2,2}}} \tag{4.83}$$

where: $Cov_{1,2}$ is the covariance between T^1 and T^2 and:

$$Cov(T^1, T^2) = E((T^1 - \mu_1) \times (T^2 - \mu_2) \tag{4.84}$$

E is the mathematical expectation and $\mu_i = E(T^i)$

The distance between T^1 and T^2 is defined as:

$$Dist_2(T^1, T^2) = 1 - (1 + Sim(T^1, T^2)) / 2 \tag{4.85}$$

Experiments performed by Pachet et al. on a database of 5000 titles show that the artist consistency may not be well enforced in the data sources. The correlation clustering generally indicates that items in a larger cluster tend to be classified according to their specific music genres, whereas the co-occurrence clustering is better suited for small clusters, indicating similarities between two titles only.

Unfortunately, the study presented by Pachet et al. was applied in a very fragmentary way, thus there is a need to perform more thorough analysis on music data in the context of searching for co-occurrence and similarity between data. Nevertheless, similar findings are applicable to address problems in extracting clusters of semantically related words. They can be found in other references, i.e. Ghazfan et al. (1996) and Klopotek (2001).

A more universal and broader approach to searching for mutual dependencies among metadata (not only textual but also numerical) is presented in the following paragraphs, basing on the Pawlak's flow graph concept (Pawlak 2004).

4.3.3 Data Mining in CDDB Database

The weakness of typical data searching techniques lays in lacking or not using any a priori knowledge concerning the queried dataset. More advanced methods assume stochastic search algorithms, which use randomized decisions while searching for solutions to a given problem. The search is based on a partial representation of data dependency expressed in terms of Bayesian networks, for example proposals formulated in papers of Ghazfan et al (1996) or Turtle et al. (1991). The process of searching is divided into 2 phases: learning and query expansion. The learning phase

stands for constructing a Bayesian network representing some of the relationships between the terms appearing in a given document collection. The query expansion phase starts when queries are issued. Queries are treated as terms that could be replaced in the process of propagating this information through the Bayesian network prepared in advance. The new terms are selected whose posterior probability is high, so that they could be added to the original query.

The abundant literature on techniques for searching data in databases describes many methods and algorithms for probabilistic searches and data mining techniques, including decision trees applications. There are no reports, however, on a successful application of any of them in representing knowledge contained in the CDDB database. Meanwhile, this problem is vital and important, because this database contains an almost complete knowledge on the sound recorded on digital disks. This knowledge, if extracted from the database, can serve for various purposes, including an efficient support for data query made by millions of users.

As a method of data mining in the CDDB database, a system application which uses logic as mathematical foundations of probability for the deterministic flow analysis in flow networks was proposed. The flow graphs are then employed as a source of decision rules providing a tool for the representation of knowledge contained in the CDDB database. The new mathematical model of flow networks underlying the decision algorithm in question was recently proposed by Pawlak (2003, 2004). The decision algorithm enables to build an efficient searching engine for the CDDB database. The proposed application is described in subsequent paragraphs.

Probabilistic and Logical Flow Networks

In the flow graphs proposed by Pawlak, flow is determined not only by Bayesian probabilistic inference rules, but also by the corresponding logical calculus which was originally proposed by Lukasiewicz. In the second case the dependencies governing flow have deterministic meaning (Pawlak 2003). The directed acyclic finite graphs are used in this context, defined as:

$$G = (N, B, \sigma) \tag{4.86}$$

$$B \subseteq N \times N \; ; \sigma : B \to \langle 0,1 \rangle \tag{4.87}$$

where:
N – a set of nodes
B – a set of directed branches

σ - a flow function

Input of $x \in N$ is the set:

$$I(x) = \{y \in N : (y,x) \in B\} \tag{4.88}$$

output of $x \in N$ is the set:

$$O(x) = \{y \in N : (x,y) \in B\} \tag{4.89}$$

Other quantities can be defined with each flow graph node, namely the inflow:

$$\sigma_+(x) = \sum_{y \in I(x)} \sigma(y,x) \tag{4.90}$$

and outflow:

$$\sigma_-(x) = \sum_{y \in O(x)} \sigma(x,y) \tag{4.91}$$

Considering the flow conservation rules, one can define the throughflow for every internal node as:

$$\sigma(x) = \sigma_+(x) = \sigma_-(x) \tag{4.92}$$

Consequently, for the whole flow graph:

$$\sigma(G) = \sigma_+(G) = \sigma_-(G), \qquad \sigma(G) = 1 \tag{4.93}$$

The factor $\sigma(x, y)$ is called **strength** of (x, y).

Above simple dependencies, known from the flow network theory, were extended by Pawlak with some new factors and a new interpretation. The definitions of these factors are following:

certainty:

$$cer(x,y) = \frac{\sigma(x,y)}{\sigma(x)} \tag{4.94}$$

coverage:

$$\mathrm{cov}(x,y) = \frac{\sigma(x,y)}{\sigma(y)} \tag{4.95}$$

The above factors have also clear logical interpretation. Provided <*x*,*y*> – represents the set of all paths leading from *x* to *y*, *certainty* has the following property:

$$cer\langle x,y\rangle = \sum_{[x...y]\in\langle x,y\rangle} cer[x...y] \tag{4.96}$$

coverage:

$$\mathrm{cov}\langle x,y\rangle = \sum_{[x...y]\in\langle x,y\rangle} \mathrm{cov}[x...y] \tag{4.97}$$

and strength fulfills the condition*:*

$$\sigma\langle x,y\rangle = \sum_{[x...y]\in\langle x,y\rangle} \sigma[x...y] \tag{4.98}$$

It is important in this context, that the flow graph branches can also be interpreted as decision rules, similarly to rough set algorithm decision rules (Pawlak 2004). That is because a decision rule *x* -> *y* can be associated with every branch (*x, y*). Consequently, a path [x_1, x_2] can be associated with a rule string: $x_1 \to x_2, x_2 \to x_3, \ldots . x_{n-1} \to x_n$ or can be represented by a single rule of the form: x^* -> x_n , where x^* replaces the string: x_1, $x_2,\ldots,x_{n-1}$

This important finding was proved in Pawlak's papers and was completed with derivation of practically useful properties, for example:

$$cer(x^*, x_n) = cer[x_1 \ldots x_n] \tag{4.99}$$

$$\mathrm{cov}(x^*, x_n) = \mathrm{cov}[x_1 \ldots x_n] \tag{4.100}$$

$$\sigma(x^*, x_n) = \sigma(x_1)\cdot cer[x_1 \ldots x_n] = \sigma(x_n)\cdot \mathrm{cov}[x_1 \ldots x_n] \tag{4.101}$$

Consequently, with every decision rule corresponding to graph branches the aforementioned coefficients are associated: flow, strength, certainty, and coverage factor. As was proved by the cited author of this applicable theory, these factors are mutually related as follows:

$$\sigma(y) = \frac{\sigma(x)\cdot cer\langle x,y\rangle}{\mathrm{cov}\langle x,y\rangle} = \frac{\sigma(x,y)}{\mathrm{cov}\langle x,y\rangle} \tag{4.102}$$

$$\sigma(x) = \frac{\sigma(y) \cdot \mathrm{cov}\langle x, y\rangle}{cer\langle x, y\rangle} = \frac{\sigma(x, y)}{cer\langle x, y\rangle} \tag{4.103}$$

If x_1 is an input and x_n is an output of graph G, then the path $[x_1...x_n]$ is complete, and the set of all decision rules associated with the complete set of the flow graph connections provides the decision algorithm determined by the flow graph.

Thus, in computer science applications of the flow graphs the concepts of probability can be replaced by factors related to flows, the latter representing data flows between nodes containing data. Knowledge on these flows and related dependencies can be stored in the form of a rule set from which the knowledge can be extracted. In contrast to many data mining algorithms described in literature, the described method is characterized by a reasonably low computational load. That is why it provides very useful means for extracting the complete knowledge on mutual relations in large data sets, thus it can be applied also as a knowledge base of an intelligent searching engine for the CDDB database.

Extracting Inference Rules from CDDB Database

Two databases in the CDDB format were selected as objects of the experiments: a local database containing metadata related to approximately 500 CD disks and the original CDDB imported from *freedb.org* website (*rev.* 20031008). At first the much smaller local database was used in order to allow experiments without engaging too much computing power for flow graph modeling. Moreover, only 5 most frequently used terms were selected as labels of node columns. These are:

- **Album title** (optional ASCII string not exceeding 256 letters)
- **Album artist** (up to 5 words separated by spaces)
- **Year** of record issuing (4 decimal digits)
- **Genre** (type of music according to the CDDB standard: Blues,...,Classical,...,Country,..., Folk,..., Jazz,..., Rock,...,Vocal). It is together 148 kinds of musical genres
- **Track title** (optional ASCII string not exceeding 256 letters)
- The term **Number** is considered a decision attribute – in the CDDB database it is represented by a unique digit/letter combination of the length equal to 8 (for example: *0a0fe010*, *6b0a4b08, etc.*).

Once the number of a record is determined, which is associated with a concrete CD, it enables retrieving all necessary metadata from the database

and rendering them by an automatic operation of filling/replacing the fields of an electronic questionnaire. The questionnaire prepared for searching CDDB databases employing knowledge extracted from flow graphs is shown in Fig. 4.30. The graph designed to represent data relations between chosen terms is illustrated in Fig. 4.31.

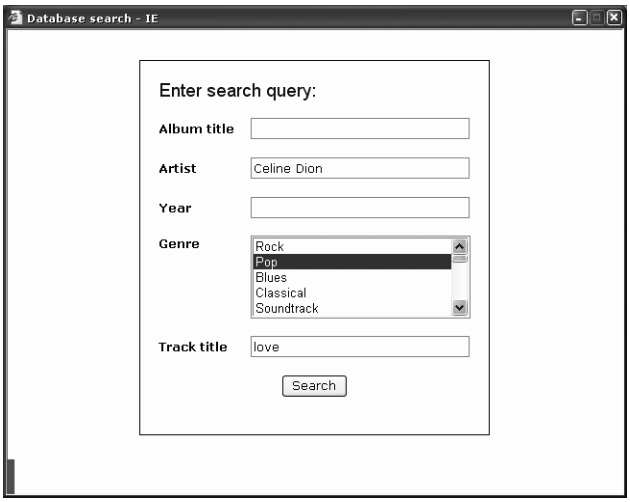


Fig. 4.30. Electronic questionnaire – front end of the CDDB searcher utilizing *a priori* knowledge about database content

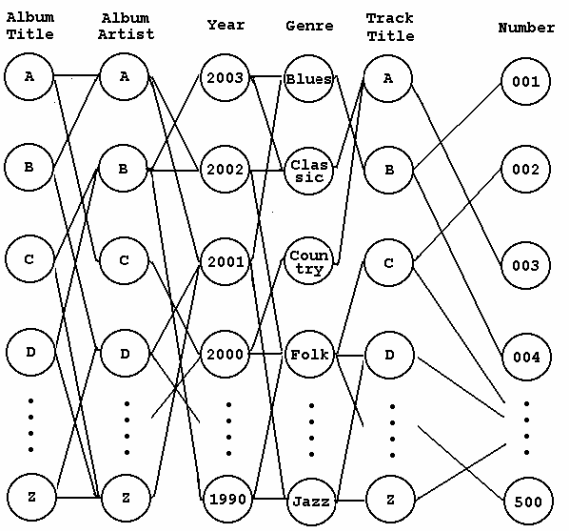


Fig. 4.31. Flow graph used to represent knowledge relevant to frequently made CDDB queries

The process of knowledge acquisition was initiated for the smaller CDDB database with analyzing first letters of the terms: ‘Album Title’, ‘Album Artist’ and ’Track Titles’. This solution was adopted because of the small size of the experimental database. Otherwise the number of paths between nodes would be too small and the problem of searching CD records will be in practice hard-defined for most objects. The above restriction does not concern the full CDDB database which contains many records of selected performers as well as many records of metadata with the same words in the fields related to album or track titles. A software implementation of the algorithm based on theoretical assumptions described earlier in the Section was prepared and implemented in a server having the following features: 2 Athlon MP 2,2 GHz processors, Windows 2000™ OS MySQL database server, Apache™ WWW server. The result of branch-related factor calculations is illustrated in Fig. 4.32.

The process of knowledge acquisition does not complete after the values of certainty, coverage and strength for each branch have been determined. The knowledge base should be prepared for queries with any reduced term set. Correspondingly, in order to determine data dependencies applicable to such cases, the graph should be simplified in advance. The knowledge base should be prepared in advance to serve such queries rather than to calculate new values of factors related to shorter paths each time a term is dropped (field left empty by the user). That is why in order to shorten the time of calculations made in response to a query, all terms are left-out consecutively, one of them at a time while the values of branch factors are calculated each time and stored. This solution lets users get a ready answer for each question almost immediately, independently of the amount of their knowledge on the CD record which is searched for. An example of a simplified flow graph is illustrated in Fig. 4.33. The dropping of the term ‘Album Artist’ node layer entails among others the following calculations:

A -> A -> 2003 ==> B -> 2003
0.0087=0.733·0.0119

C -> B -> 2002 ==> C -> 2002
0.0012=0.1875·0.0063

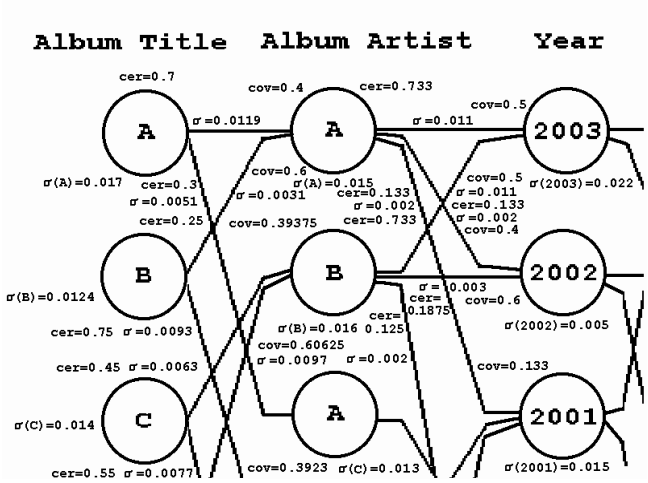


Fig. 4.32. Fragment of flow graph with marked values of **certainty**, **coverage** and **strength** calculated for branches

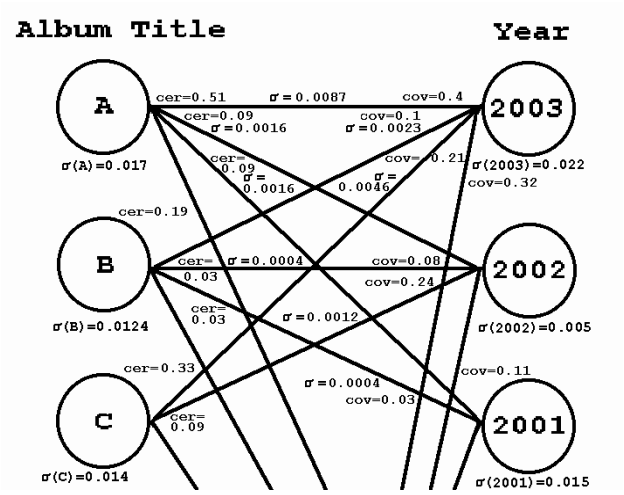


Fig. 4.33. Simplified flow graph (from Fig. 4.32) after leaving-out the term: 'Album artist'

As said before, decision rules can be derived from flow graphs. Correspondingly, the following sample inference rules can be obtained from the graph showed in Fig. 4.32, whose fragment is depicted in Fig. 4.33:

If *Album Title=B* **and** *Album Artist=A* **and** *Year=2003* **and** *Genre=genre_value* **and** *Track Title=track_title_value* **then** *Number=number_value*

If *Album Title=C* **and** *Album Artist=B* **and** *Year=2002* **and** *Genre=genre_value* **and** *Track Title=track_title_value* **then** *Number=number_value*

The values of: *genre_value, track_title_value and number_value* can be determined from the parts of the graph that are not covered by the figure (for caption resolution limitations). If the user did not provide the *Album Artist* value, the direct data flows from the nodes *Album Title* to the nodes *Year* and can be analyzed as in Fig. 4.33. The inference rules are shorter in this case and adequate values of **certainty**, **coverage** and **strength** are adopted. For example the value of the rule strength associated with the paths determined by the node values: *Album Title=B -> Album Artist=A* (as in Fig. 4.32) equal to σ=0.0031 and σ=0.0011 are replaced by the new value of σ=0.0023 associated with the path: *Album Title=B -> Year=2003*. The shortened rules corresponding to the previous examples given above are as follows:

If *Album Title=B* **and** *Year=2003* **and** *Genre=genre_value* **and** *Track Title=track_title_value* **then** *Number=nember_value*

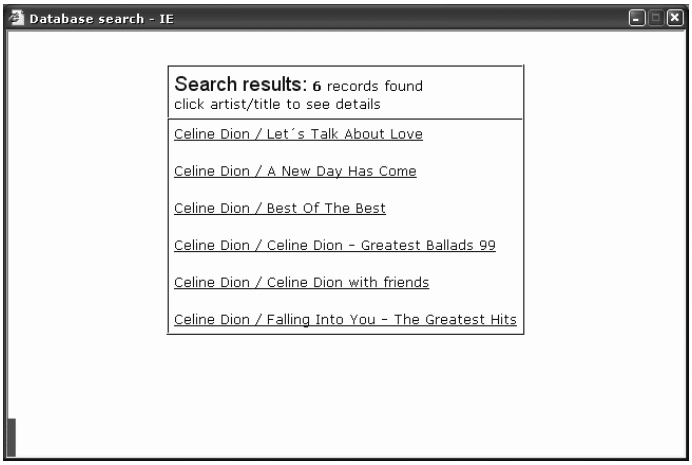
If *Album Title=C* **and** *Year=2002* **and** *Genre=genre_value* **and** *Track Title=track_title_value* **then** *Number=number_value*

The latter inference rules may adopt the same decision attribute (the number of the same CD record), however the rule strength (σ value) can be different in this case. The rule strength is a decisive factor for ordering the results of the search in the database. The principle of ordering matches is simple: the bigger the rule strength value is, the higher is the position of the CD record determined by the rule in the ordered rank of matches. This principle enables a descendant ordering of queried CDs. It bases on the rules derived from the analysis of optimal data flow in the graphs which represent the knowledge on CD records.

It is interesting in this context that the attributes in the decision rules can be reordered as long as a route between the same nodes consisting of the same branches is covered by the rule. The rule can be also reversed or the decision attribute can be swapped with any former conditional attribute. This feature of the decision system results from the principles of *modus ponens* and *modus tollens* valid for logical reasoning made on the basis of

flow graphs. Correspondingly, the user of the searching engine who knows the number of a CD, can find information about the remaining metadata based on this decision system. It is also interesting that in this context the detailed - practically impossible for a user to memorize - number of beginning packets of tracks might not be necessary, thus the CD album metadata can be searched for effectively without an access to its physical copy. The panels of the constructed searching engine utilized for the presentation of the search results are gathered in Fig. 4.34.

a.



b.

Fig. 4.34. Panels of the intelligent CDDB database searcher containing the results of a query (from Fig. 4.32): (a) result of matches rendering; (b) retrieved metadata

An application of the knowledge extraction algorithm to the CDDB case is practically justified provided it is possible to complete all computing tasks on a typical server (full set of inference rule derivation). This demand is entirely fulfilled in the case of flow graphs application.

Apart from the experiments on semantic music classification, an attempt to use parameters from the MPEG-7 standard for a musical style classification was carried out. A set of 162 musical pieces was used in the experiment. Each of them was categorized to be classical music, jazz or rock.

Approximately 15 one-second samples, starting one minute from the beginning of a piece, were extracted from every piece. Eight randomly chosen samples from every piece were added to the training set, depending on the classification algorithm. Other samples were included in the test set. Parameters from the MPEG-7 standard applicable for such an analysis were included in the feature vector. Therefore, the feature vector consisted of only Audio Spectrum Descriptors. Results of the musical style classification are shown in Tables 4.37 and 4.38. Both NN and rough set-based classifiers were used. In the latter case, the RSES system was used (Bazan and Szczuka 2001; Bazan et al 2002). The results obtained in musical style classification are lower by approximately 10% than the results of musical instrument identification. It is believed that extending the feature vector by the specialized rhythm parameters would improve the classification effectiveness significantly.

The content of the feature vector used in the experiments was as follows:

$\{ASE_1, ASE_2, ASE_3, ASE_4, ASE_5, ASE_6, ASE_7, ASE_8, ASE_9, ASE_{10}, ASE_{11}, ASE_{12}, ASE_{13}, ASE_{14}, ASE_{15}, ASE_{16}, ASE_{17}, ASE_{18}, ASE_{19}, ASE_{20}, ASE_{21}, ASE_{22}, ASE_{23}, ASE_{24}, ASE_{25}, ASE_{26}, ASE, ASEv, ASC, ASCv, ASS, ASSv, SFM, SFMv\}$.

Table 4.37. Effectiveness of musical style classification by means of NN

Genre	jazz	classical	rock	No. of obj.	Accuracy	Coverage
jazz	406	68	31	505	0.804	1
classical	52	355	5	412	0.862	1
rock	29	13	128	176	0.727	1

Table 4.38. Effectiveness of musical style classification by means of rough sets

Genre	jazz	classical	rock	No. of obj.	Accuracy	Coverage
jazz	426	101	46	573	0.743	1
classical	58	411	5	474	0.867	1
rock	32	10	264	306	0.863	1

Nearly the same effectiveness was obtained by both the neural network and the rough set-based classification system. In the latter case, such an analysis was slightly better suited for classification tasks than NN. It can be noticed that the difference between the results obtained by these algorithms is nearly negligible, however, the results are not yet satisfactory.

In more practical applications, a user is satisfied with a classification if two factors are fulfilled. First, the system should return results of music identification in a fraction of second, secondly the correct result should appear at the first five positions of the list returned by the system.

The database with 4469 fragments of music excerpts of various genres was once again used in experiments. From the whole database 60 pieces were randomly extracted and parametrized. To satisfy time conditioning, only two parameters in feature vectors were used, namely *AudioPower* and *AudioSpectrumFlatness*. Since the values of the first parameter are 10 times larger than the values of *AudioSpectrumFlatness*, thus the values of the *AudioPower* were used with weights of 0.1 and 0.01 (see Fig. 4.35). These parameters were calculated for 60s of the music excerpt, only.

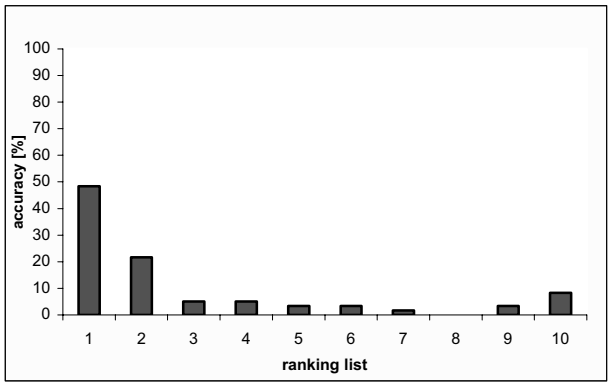


Fig. 4.35. Results returned by the system (weight 0.01 applied to *AudioSpectrumFlatness*)

As seen from Fig. 4.35, the system identified the correct music excerpt with the accuracy of 48.3%. It should be stressed that in such a case only two parameters were contained in the feature vector. The analysis of the remaining four places of the ranking list shows that the accuracy increases to 81.6% (49 correct guesses), and testing the system with the weight of 0.1 applied to the *AudioSpectrumFlatness* parameter, has a different effect on the results – namely the first place in the ranking list is returned with 38.33% accuracy (see Fig. 4.36). The system was prepared within one of the M.Sc. Thesis of the Multimedia Systems Department, and is installed

on the Department server, however all interfaces are in Polish, thus there are not shown here (Prylowski 2004).

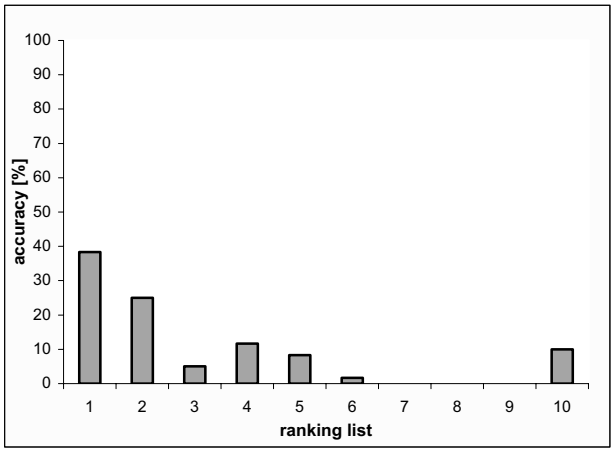


Fig. 4.36. Results returned by the system (weight 0.1 applied to *AudioSpectrumFlatness*)

The results indicate that even in very demanding conditions, a correct classification is not only possible, but also satisfying. It should be mentioned that, if a two-fold search of the database is applied – namely when high level parameters (semantic) are added – then the system returns adequate list of music excerpts with 100% accuracy.

References

Abed-Meraim K, Belouchrani A, Hua Y (1996) Blind Identification of a Linear-Quadratic Mixture. In: Proc IEEE Intl Conf on Acoust, Speech and Sig Proc 5

McAdams S (1989) Concurrent sound segregation. I: Effects of frequency modulation coherence. J Acoust Soc Am 86: 2148-2159

McAdams S, Winsberg S, Donnadieu S, De Soete G, Krimphoff J (1995) Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes. Psychological Research 58, pp 177-192

Amari S (1999) ICA of temporally correlated signals-learning algorithm. In: Proc ICA and Sig

Amari S, Cichocki A, Yang HH (1996) A New Learning Algorithm for Blind Signal Separation. Advances in Neural Info Processing Systems 8: 1–7

McAulay RJ, Quatieri TF (1990) Pitch estimation and voicing detection based on sinusoidal speech model. In: Proc ICASSP 1, pp 249-252

Baesa-Yates R, Perleberg C (1992) Fast and practical approximate string matching. In: Proc of Combinatorial Pattern Matching, pp 185-192, Springer, USA

Bainbridge D, Cunningham SJ, Downie JS (2003) How People Describe Their Music Information Needs: A grounded Theory Analysis of Music Queries. In: Proc Fourth International Conference on Music Information Retrieval (ISMIR), Baltimore

Balan R, Rosca J, Rickard S (2001) Robustness of Parametric Source Demixing in Echoic Enviroments. Third Int Symposium on Independent Component Analysis and Blind Source Separation, San Diego

Barucha JJ, Todd PM (1991) Modeling the Perception of Tonal Structure with Neural Nets. In: Todd PM, Loy DG (eds) Music and Connectionism. The MIT Press, Cambridge Massachusetts London, pp 128-137

Bazan J, Szczuka M (2001) RSES and RSESlib - A Collection of Tools for Rough Set Computations. In: Proc of RSCTC'2000, LNAI 2005. Springer Verlag, Berlin

Bazan J, Szczuka M, Wroblewski J (2002) A New Version of Rough Set Exploration System. In: Alpigini JJ (ed) Proc RSCTC, LNAI 2475. Springer Verlag, Heidelberg, Berlin, pp 397-404

Bell A, Sejnowski T (1995a) An Information - Maximisation Approach to Blind Separation and Blind Deconvolution. Neural Computation 7: 1129-1159

Bell A, Sejnowski T (1995b) Fast blind separation based on information theory. In: Proc Intern Symp on Nonlinear Theory and Applications, Las Vegas, pp 43-47

Belouchrani A, Abed-Meraim K, Cardoso J-F, Moulines E (1997) A blind source separation technique using second order statistics. IEEE Trans on Signal Processing

Braut Ch (1994) The Musician's Guide to MIDI. SYBEX, San Francisco

Bregman AS (1990) Auditory Scene Analysis. MIT press

Bullivant R (1980) Fugue, The New Grove Dictionary of Music and Musicians. Macmillan Publishers Limited, London 7, pp 9-21

Byrd D, Crawford T (2002) Problems of music information retrieval in the real world. Information Processing and Management 38, Issue 2: 249-272

Cardoso J (1992) Iterative Techniques for Blind Source Separation Using Only Fourth-Order Cumulants. In: Proc EUSIPCO-92, Brussels, pp 739-742

Casey MA (2001) Separation of mixed audio sources by independent subspace analysis. International Computer Music Conference (ICMC)

Cendrowski J (2005) Examination of Separation Algorithm Effectiveness. MSc Thesis, Multimedia Systems Department, Gdansk University of Technology (*in Polish*), Kostek B (supervisor)

Chan DCB (1997) Blind Signal Separation. PhD thesis, Trinity College, Cambridge University

de Cheveigné A (1993) Separation of concurrent harmonic sounds: frequency estimation and a time domain cancellation model of auditory processing. J Acoust Soc Am

Chin F, Wu S (1992) An Efficient Algorithm for Rhythm-finding. Computer Music Journal, Massachusettes Institute of Technology 16, No 2, pp 35–44

Chmielewski MR, Grzymala-Busse JW (1994) Global Discretization of Continuous Attributes as Preprocessing for Machine Learning. In: Proc 3rd International Workshop on Rough Sets and Soft Computing, San Jose

Choi S, Cichocki A, Zhang L, Amari S (2001) Approximate Maximum Likelihood Source Separation Using the Natural Gradient. In: Proc IEEE Workshop on Signal Processing Advances in Vireless Communications, Taoyuan, pp 235-238

Cosi P, De Poli G, Prandoni P (1994b) Timbre characterization with Mel-Cepstrum and Neural Nets. In: Proc of the 1994 ICMC, pp 42-45

Czyzewski A (2003) Automatic Identification of Sound Source Position Employing Neural Networks and Rough Sets. Pattern Recognition Letters 24: 921 - 933

Czyzewski A, Kostek B (2004) Musical Metadata Retrieval with Flow Graphs, in Rough Sets and Current Trends in Computing. In: Proc 4th International Conf Rough Sets Current Trends in Computing, Lecture Notes in Artificial Intelligence, LNAI 3066, Springer Verlag, Berlin, Heidelberg, New York, pp 691-698

Czyzewski A, Kostek B, Krolikowski R (2001) Neural Networks Applied to Sound Source Localization. In: Proc 110th Audio Eng Soc Conv, Amsterdam

Czyzewski A, Kostek B, Lasecki J (1998) Microphone Array for Improving Speech Intelligibility. In: Proc 20 Tonmeistertagung, International Convention on Sound Design, Stadthalle Karlsruhe, Germany, 428-434

Czyzewski A, Szczerba M (2002) Pitch Estimation Enhancement Employing Neural Network- Based Music Prediction. In: Proc IASTED Intern Conference, Artificial Intelligence and Soft Computing, pp 413-418, Banff, Canada

Czyzewski A, Szczerba M, Kostek B (2004) Musical Phrase Representation and Recognition by Means of Neural Networks and Rough Sets. Rough Set Theory and Applications (RSTA) 1, pp 259-284, Advances in Rough Sets, Subseries of Springer-Verlag Lecture Notes in Computer Sciences, LNCS 3100, Trans on Rough Sets, Grzymala-Busse JW, Kostek B, Swiniarski RW, Szczuka M (eds)

Deller JR, Proakis JC, Hansen JHL (1993) Discrete-Time Processing of Speech Signals. Macmillan

Desain P (1992) A (de)composable theory of rhythm perception. Music Perception 9: 439-454

Desain P, Honing H (1991) The Quantization of Musical Time: A Connectionist Approach. In: Todd PM, Loy DG (eds) Music and Connectionism. The MIT Press, Cambridge Massachusetts London, pp 150-169

Desain P, Honing H (1997) Music, Mind, Machine: Computational Modeling of Temporal Structure in Musical Knowledge and Music Cognition. Informatie, 39: 48-53

Dixon S (2001) Automatic Extraction of Tempo and Beat from Expressive Performances. Journal of New Music Research, Swets & Zeitlinger 30, No 1: 39-58

Downie JS (1999) Music retrieval as text retrieval: simple yet effective. In: Proc of SIGIR '99, 22nd Int Conf on Research and Development in Information Retrieval, ACM, New York, pp 297-8

Downie JS (2003) Music information retrieval. In: Cronin B (ed) Annual Review of Information Science and Technology 37. Medrofd, NJ, Information Today, pp 295-340. Available from URL: http://music-ir.org/downie_mir_arist37.pdf

Downie JS (2003) Toward the Scientific Evaluation of Music Information Retrieval Systems. Fourth International Conference on Music Information Retrieval (ISMIR), Baltimore

Duvall KM (1983) Signal cancellation in adaptive antennas: the phenomenon and a remedy. Stanford Univ

Dziubinski M (2005) Evaluation of Musical Instrument Sound Separation Method Effectiveness in Polyphonic Recordings by Means of Soft Computing Methods. Ph.D. thesis, Multimedia Systems Department, Gdansk University of Technology (in preparation)

Dziubinski M, Dalka P, Kostek B (2005) Estimation of Musical Sound Separation Algorithm Effectiveness Employing Neural Networks. J Intelligent Information Systems, Special Issue on Intelligent Multimedia Applications 24, No 2: 1333-157 (*in print*)

Ellis DPW (1996) Prediction-driven computational auditory scene analysis. Ph.D. thesis, Massachusetts Institute of Technology

Feulner J, Hörnel D (1994) MELONET: Neural Networks that Learn Harmony-Based Melodic Variations. In: Proc International Computer Music Conference, San Francisco: International Computer Music Association, pp 121-124

Foote J (1997) Content-Based Retrieval of Music and Audio URL: http://www.fxpal.com/people/foote/papers/spie97.pdf – Content-Based Retrieval of Music and Audio – TreeQ System

Frost OL (1972) Adaptive least-squares optimization subject to linear equality constraints. Stanford Univ

Ghazfan D, Indrawan M, Srinivasan B (1996) Towards meaningful Bayesian belief networks. In: Proc IPMU, pp 841-846

Ghias A, Logan H, Chamberlin D Smith BC (1995) Query by humming. In: Electronic Proc of ACM Multimedia 95, San Francisco URL: http://www.acm.org/pubs/articles/proceedings/multimedia/217279/p231-ghias/p231-ghias.htm

Griffiths LJ, Jim CW (1982) An alternative approach to linearly constrained adaptive beamforming. IEEE Trans Antennas and Propagation 30: 27-34

Greenberg JE et al (1992) Evaluation of an Adaptive Beamforming Method for Hearing Aids. J Acoust Soc Am 91 (3): 1662 – 1676

Grey JM (1977) Multidimensional perceptual scaling of musical timbres. J Acoust Soc Am 61: 1270-1277

Hawley M, The personal orchestra, Computing Systems, 1990 3, No 2, pp 289-329

Hawley ML, Litovsky RY, Colburn H S (1999) Speech intelligibility and localization in multi-source environment. J Acoust Soc Am 105 (6): 3436-3448

Herrera P, Amatriain X, Battle E, Serra X (2000) Towards Instrument Segmentation for Music Content Description: a Critical Review of Instrument Classification Techniques. In: Proc Intern Symposium on Music Information Retrieval, ISMIR 2000, URL: http://ismir2000.indiana.edu/2000

Hörnel D (1997) MELONET I: Neural Nets for Inventing Baroque-Style Chorale Variations. In: Jordan MI, Kearns MJ, Solla SA (eds) Advances in Neural Information Processing 10 (NIPS 10), MIT Press

Jung A (2001) An introduction to a new data analysis tool: Independent Component Analysis. In: Proc of Workshop GK Nonlinearity, Regensburg

Jung A, Kaiser A (2003) Considering temporal structures in Independent Component Analysis. In: Proc ICA 2003

Karhunen J (1996) Neural Approaches to Independent Component Analysis and Source Separation. In: Proc 4th European Symposium on Artificial Neural Networks ESANN'96, Bruges, pp 249-266

Karjalainen M, Tolonen T (1999) Multi-pitch and periodicity analysis model for sound separation and auditory scene analysis. In: Proc ICASSP, Phoenix

Kates JM, Weiss MR (1996) A comparison of hearing-aid array-processing techniques. J Acoust Soc Am 99 (5): 3138-3148

Klapuri A (1998) Automatic Transcription Of Music. Tampere University of Technology

Klapuri A, Virtanen T, Holm J (2000) Robust multipitch estimation for the analysis and manipulation of poliphonic musical signals. In: Proc COST-G6 Conference on Digital Audio Effects, Verona

Klopotek MA (2001) Intelligent Search Engines (*in Polish*). Scientific Press EXIT, Warsaw

Komorowski J, Pawlak Z, Polkowski L, Skowron A (1998) Rough Sets: A Tutorial. In: Pal SK, Skowron A (eds) Rough Fuzzy Hybridization: A New Trend in Decision-Making. Springer-Verlag, pp 3-98

Kostek B (1995) Computer Based Recognition of Musical Phrases Using the Rough Set Approach. 2nd Annual Joint Conference on Inform Sciences, pp 401-404

Kostek B (1998) Computer-Based Recognition of Musical Phrases Using the Rough-Set Approach. J Information Sciences 104: 15-30

Kostek B (1999) Soft Computing in Acoustics, Applications of Neural Networks, Fuzzy Logic and Rough Sets to Musical Acoustics, Studies in Fuzziness and Soft Computing. Physica Verlag, Heidelberg New York

Kostek B (2004a), Application of soft computing to automatic music information retrieval. J American Society for Information Science and Technology 55, No 12: 1108-1116

Kostek B (2004b) Musical Instrument Classification and Duet Analysis Employing Music Information Retrieval Techniques. Proc of the IEEE 92, No 4: 712-729

Kostek B, Czyzewski A, Lasecki J (1999) Spatial filtration of sound for multimedia systems. In: Proc IEEE 3rd Workshop on Multimedia Signal Processing, Copenhagen, Denmark

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

Kostek B, Czyzewski A (2004) Processing of Musical Metadata Employing Pawlak's Flow Graphs. In: Rough Set Theory and Applications (RSTA), Advances in Rough Sets, Subseries of Springer-Verlag Lecture Notes in Computer Sciences, LNCS 3100, Trans on Rough Sets 1, Grzymala-Busse JW, Kostek B, Swiniarski RW, Szczuka M (eds), pp 285-305

Kostek B, Dziubiński M, Zwan P (2002b), Further developments of methods for searching optimum musical and rhythmic feature vectors. In: Proc 21st Audio Engineering Soc Conference, St. Petersburg, Russia

Kostek B, Dziubinski M, Dalka P (2004) Comparison of Effectiveness of Musical Sound Separation Algorithms Employing Neural Networks. In: Proc 117 Engineering Soc Convention, Preprint No Z6-4, San Francisco

Kostek B, Szczerba M (1996a) MIDI Database for the Automatic Recognition of Musical Phrases. In: Proc 100th AES Convention, preprint 4169, Copenhagen. J Audio Eng Soc, (Abstr) 44: 10

Kostek B, Szczerba M (1996b) Parametric Representation of Musical Phrases. In: Proc 101st AES Convention, preprint 4337, Los Angeles (1996). J Audio Eng Soc, (Abstr) 44, No 12: 1158

Kostek B, Wojcik J (2004) Forming and Ranking Musical Rhythm Hypotheses. In: Proc of Knowledge-Based Intelligent Information & Engineering Systems, Lecture Notes in Artificial Intelligence Wellington, New Zealand

Kostek B, Wojcik J (2005) Machine Learning System for Estimation Rhythmic Salience of Sounds. KES Journal (*in print*)

Kostek B, Wojcik J, Holonowicz P (2005) Estimation the Rhythmic Salience of Sound with Association Rules and Neural Network. Intelligent Information Systems 2005, Springer Verlag

Kostek B, Zwan P, Dziubinski M (2002a) Statistical Analysis of Musical Sound Features Derived from Wavelet Representation. In: Proc 112th Audio Engineering Society Conference,Munich, Germany

Krimphoff J, McAdams S, Winsberg S (1994) Caracterisation du timbre des sons complexes. II Analyses acoustiques et quantification psychophysique. J Phys 4: 625-628

Kusuma J (2000) Parametric frequency estimation: ESPRIT and MUSIC. MITTutorial, URL: http://web.mit.edu/kusuma/www/Papers/parametric.pdf

Langendijk EHA, Bronkhorst AW (2000) Fidelity of three-dimensional-sound reproduction using a virtual auditory display. J Acoust Soc Am 107(1): 528-537

Lasecki J, Czyzewski A, Kostek B (1998) Neural Network-Based Algorithm For the Improvement of Speech Intelligibility. In: Proc of 20th International Convention on Sound Design, pp 428-434, Karlsruhe

Lasecki J, Czyzewski A (1999) Neural Network-Based Spatial Filtration Algorithm For 2-Microphone Array. In: Proc Acoust Soc Am Meeting, Berlin, Germany

Lasecki J, Kostek B, Czyzewski A (1999) Neural Network-Based Spatial Filtration of Sound. In: Proc 106th Audio Eng Soc Convention, Preprint No 4918 (J4), Munich, Germany

Laurenti N, de Poli G (2000) A method for spectrum separation and envelope estimation of the residual in spectrum modelling of musical sound. In: Proc COST G6 Conf on Digital Audio Effects, DAFX-00. Verona

Liebermeister W (2002) Linear modes of gene expression determined by independent component analysis. Bioinformatics 18: 51-60

Lerdahl F, Jackendoff R (1983) A generative theory of tonal music. MIT Press, Cambridge, MA

Lindsay AT, Herre J (2001) MPEG-7 and MPEG-7 Audio – An Overview. J Audio Eng Soc 49: 589-594

Mannila H (1996) Data mining: machine learning, statistics, and databases. In: Proc Eighth International Conference on Scientific and Statistical Database Management. IEEE Comput Soc Press, Los Alamitos, CA, USA, pp 2-9

Mansour A, Barros A, Ohnishi N (2000) Blind separation of sources: Methods, Assumptions and applications. IEICE Trans Fundamentals E83-A, No 8

Mapp P et al (1999) Improvements in Teleconferencing Sound Quality and Gain before Feedback through the use of DML Technology. In: Proc 137th Acoust Soc Am Meeting, Berlin, Germany

Masuda-Katsuse I (2001) A new method for speech recognition in the presence of non stationary, unpredictable and high level noise. In: Proc Euro-Speech 2001, pp 1119-1122

Meddis R, Hewitt MJ (1991) Virtual pitch and phase-sensitivity of a computer model of the auditory periphery. J Acoust Soc Am 89: 2866-2882

Mellinger DK (1991) Event Formation and Separation in Musical Sound. Ph.D. thesis, Center for Computer Research in Music and Acoustics, Stanford University

Merks I et al (1999) Array technology for binaural hearing-aids applications. In: Proc Acoust Soc Am Meeting, Berlin, Germany

Moradi H, Grzymala-Busse JW, Roberts JA (1998) Entropy of English Text: Experiments with Humans and a Machine Learning System Based on Rough Sets. J Information Sciences 104: 31-47

Mozer MC (1991) Connectionist Music Composition Based on Melodic, Stylistic, and Psychophysical Constraints. In: Todd PM, Loy DG (eds) Music and Connectionism. The MIT Press, Cambridge Massachusetts London, pp 195-211

McNab RJ, Smith LA, Witten IH (1996) Signal processing for melody transcription. James Cook Univ Australian Computer Science Communications, Australia 18, No 1, pp 301-307

McNab RJ, Smith LA, Witten IH, Henderson CL, Cunningham SJ (1996) Towards the digital music library: tune retrieval from acoustic input. In: Proc 1st ACM Int Conf on Digital Libraries, ACM, New York, pp 11-18

Nguyen HS (1998) Discretization Problem for Rough Sets Methods. In: Proc RSCTC'98, Lecture Notes in Artificial Intelligence, No 1424, Springer Verlag, Rough Sets and Current Trends in Computing, Polkowski L, Skowron A (eds), pp 545-552

Oppenheim AV, Lin JS (1981) The importance of phase in signals. Proc of the IEEE 69: 529-541

Øhrm A (1999) Discernibility and Rough Sets in Medicine: Tools and Applications. Ph.D. Thesis, Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim (NTNU Report 1999: 133, IDI Report)

Pachet F, Westermann G, Laigre D (2002) Musical Data Mining for Electronic Music Distribution. WedelMusic, Firenze

Pal SK, Polkowski L, Skowron A (2004) Rough-Neural Computing. Techniques for Computing with Words. Springer Verlag, Berlin Heidelberg New York

Parncutt RA (1994) A perceptual model of pulse salience and metrical accent in musical rhythms. Music Perception 11(4): 409-464

Pawlak Z (1982) Rough Sets. International J Computer and Information Sciences 11

Pawlak Z (2003) Probability, Truth and Flow Graph. Electronic Notes in Theoretical Computer Science 82, International Workshop on Rough Sets in Knowledge Discovery and Soft Computing, Satellite event of ETAPS 2003, Elsevier, Warsaw

Pawlak Z (2004) Elementary Rough Set Granules: Towards a Rough Set Processor. In: Pal SK, Polkowski L, Skowron A (eds) Rough-Neural Computing. Techniques for Computing with Words. Springer Verlag, Berlin Heidelberg New York, pp 5-13

Pfeiffer S, Fischer S, Effelsberg W (1996) Automatic audio content analysis. In: Proc ACM Multimedia 96, New York, pp 21-30

Pohlman K (1992) The Compact Disk Handbook. A-R editions

De Poli G, Prandoni P (1997) Sonological models for timbre characterization. J New Music Research 26: 170-197

Povel DJ and Essens P (1985) Perception of temporal patterns. Music Perception 2(4): 411-440

Proakis JG, Manolakis DG (1999) Digital Signal Processing. Principles, Algorithms and Applications. Third Edition, Prentice Hall International

Prylowski L (2004) Web-based Music Retrieval Application. MSc Thesis, Multimedia Systems Department, Gdansk University of Technology (*in Polish*), Kostek B (supervisor)

Rabiner L, Cheng MJ, Rosenberg AE, Gonegal C (1976) A Comparative Performance Study of Several Pitch Detection Algorithms. IEEE Trans Acoustic, Speech, Signal Processing 24: 399-418

Repp BH (1996) Patterns of note asynchronies in expressive piano performance. J Acoust Soc Am 100: 3917-3932

Rosenthal, DF (1992a) Emulation of human rhythm perception. Computer Music Journal 16, No: 64-76

Rosenthal DF (1992b) Machine Rhythm: Computer Emulation of Human Rhythm Perception. PhD thesis, MIT Media Lab, Cambridge, MASS

Satayarak P, Rawiwan P, Supanakoon P, Chamchoy M, Promwong S, Tangtisanon P (2002) The Achievable Performance of Unitary-ESPRIT Algorithm for DOA Estimation. In: Proc The 2002 International Technical Conference

on Circuits/Systems, Computers and Communications, ITC-CSCC 2002, pp 1578-1581

Serra X (1997) Musical Sound Modelling with Sinusoids plus Noise. Musical Signal Processing. Swets & Zeitlinger Publishers

Skowron A, Nguyen SH (1995) Quantization of Real Value Attributes, Rough Set and Boolean Approach. (ICS Research Report 11/95, Warsaw University of Technology)

Slaney M, Naar D, Lyon RF (1994) Auditory model inversion for sound separation. In: Proc ICASSP

Soede W, Bilsen FA, Berkhout AJ (1993) Assessment of a directional microphone array for hearing impaired listeners. J Acoust Soc Am 94 (2): 799-808

Swindlehurst A, Ottersten B, Roy R, Kailath T (1992) Multiple Invariance ESPRIT. IEEE Trans on Signal Proc 40, 4: 867-881

Swindlehurst A L, Stoica P and Jansson M (2001) Exploiting Arrays with Multiple Invariances Using MUSIC and MODE. IEEE Trans on Signal Processing 49, 11: 2511-2521

Swiniarski R (2001) Rough sets methods in feature reduction and classification. Int J Applied Math Comp Sci 11: 565-582

Szczerba M (1999) Recognition and Prediction of Music: A Machine Learning Approach. In: Proc of 106th AES Convention, Munich

Szczerba M (2002) Recognition and Prediction of Music, a Machine Learning Approach. PhD Thesis, Multimedia Systems Department, Gdansk University of Technology (*in Polish*), Czyzewski A (supervisor)

Tanguiane AS (1991) Artificial Perception and Music Recognition. Lecture Notes in Artificial Intelligence No 746, Springer Verlag

The New Grove Dictionary of Music and Musicians (1980). Macmillan Publishers Limited, Sadie (ed) 1, London, pp 774-877

Todd PM (1991) A Connectionist Approach to Algorithmic Composition. In: Todd PM, Loy DG (eds) Music and Connectionism. The MIT Press, Cambridge Massachusetts London, pp 173-194

Toiviainen P, Tervaniemi M, Louhivuori J, Saher M, Huotilainen M, and Näätänen R (1998) Timbre Similarity: Convergence of Neural, Behavioral, and Computational Approaches. Music Perception 16: 223-241

Torkkola K (1996) Blind Separation of Delayed Sources Based on Information Maximization. In: Proc ICASSP96, pp 3509-3512

Torkkola K (1999) Blind Separation for Audio Signals - Are We There Yet? In: Proc ICA'99, pp 239-244

Tseng Y H (1998) Multilingual Keyword Extraction for Term Suggestion. In: Proc 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, pp 377-378

Tseng YH (1999) Content-based retrieval for music collections. In: Proc of SIGIR '99, 22nd International Conference on Research and Development in Information Retrieval, New York, pp 176-82

Turtle HR, Croft WB (1991) Evaluation of an inference network-based retrieval model. In: ACM Trans on Information Systems 9, pp 187-222

Valpola H, Oja E, Ilin A, Honkela A, Karhunen J (2003) Nonlinear Blind Source Separation by Variational Bayesian Learning. IEICE Trans (Japan) E86-A, No 3: 532-541.

Veen BD, Buckley KM (1988) Beamforming: A Versatile Approach to Spatial Filtering. IEEE Acoust, Speech and Sig Proc Mag 5(2): 4-24

Viste H, Evangelista G (2002) An extension for source separation techniques avoiding beats. In: Proc of the 5th Int Conference on Digital Audio Effects

Walmsley PJ (2000) Signal Separation of Musical Instruments - Simulation-based methods for musical signal decomposition and transcription. Ph.D. thesis, University of Cambridge

Weintraub M (1986) A computational model for separating two simultaneous talkers. In: Proc ICASSP

Wessel D (1979) Timbre space as a musical control structure. J Computer Music 3: 45-52

Westner A, Bove V (1999) Applying Blind Source Separation and Deconvolution to Real-World Acoustic Environments. In: Proc 106th Audio Eng Soc Convention, Preprint No 4918 (J4), Munich, Germany, Preprint No 4955

Wojcik J, Kostek B, Holonowicz P (2004) Neural Network and Data Mining Approaches to Estimate the Salience of Musical Sound in Melody. In: Proc of VDT International Audio Convention, Tonmeistertagung Leipzig

Wojcik J, Kostek B, (2004) Intelligent Methods for Musical Rhythm Finding Systems. Chapter in Intelligent Technologies for Inconsistent Processing, Ngoc Thanh Nguyen (ed), International Series on Advanced Intelligence 10, pp 187-202

Wu S, Manber U (1992) Fast text searching allowing errors, Communications of the ACM 35, No 10: 83-91

Yellin D, Weinstein E (1994) Criteria for multichannel signal separation. In: IEEE Trans on Signal Processing

Zell A (2002) SNNS – Stuttgart Neural Network Simulator User Manual, Ver 4.1.

URL: http://ismir2001.ismir.net/posters/jang.pdf – MIRACLE: A Music Information Retrieval System with Clustered Computing Engines

URL: http://ismir2002.ismir.net/proceedings/02-FP06-2.pdf – CubyHum: A Fully Operational Query by Humming System

URL: http://mirsystems.info/ (2005) A Survey of Music Information Retrieval Systems, Rainer Typke home page

URL: http://omras.dcs.kcl.ac.uk/Full_desc.html – OMRAS System

URL: http://theremin.music.uiowa.edu/MIS.html - Musical Instrument Sounds, Iowa University

URL: http://www.audiograbber.com-us.net/

URL: http://www.cebit2003.fraunhofer.de/servlet/is/4107/ – Fraunhofer IIS Query by Humming – MelodieSuchmaschine System

URL: http://www.cs.ubc.ca/~hoos/Publ/ismir01.pdf – GUIDO/MIR – an Experimental Musical Information Retrieval System based on GUIDO Music Notation

URL: http://www.cs.uu.nl/people/rtypke/transdist_ismir2003.pdf – Using Transportation Distances for Measuring Melodic Similarity – Orpheus System
URL: http://www.darmstadt.gmd.de/mobile/MPEG7/
URL: http://www.dlib.org/dlib/may97/meldex/05witten.html – MELDEX System.
URL: http://www.freedb.org
URL: http://www.gracenote.com
URL: http://www.id3.org
URL: http://www.iis.fraunhofer.de/amm/download/qbh.pdf – Query by Humming Melody Recognition System – MelodieSuchmaschine System
URL: http://www.informatik.tu-darmstadt.de/AFS/GUIDO/
URL: http://www.ismir.net
URL: http://www.jsbach.org webpage BWV - *Bach Werke Verzeichnis*
URL: http://www.meta-labs.com/mpeg-7-aud
URL: http://www.prs.net Classical MIDI Archives
URL: http://www.semanticweb.org/
URL: http://www.w3.org/2001/sw/
URL: http://213.133.111.178/Rntt/tuneserver_tochi2001.pdf – Tuneserver System - An Interface for Melody Input

5 COGNITIVE PROCESSING IN ACOUSTICS

5.1 Fuzzy Sets and Fuzzy Logic

The idea of vagueness (contrary to bi-valent logic) appeared at the end of the 19th century, and was formally applied to the field of logic in 1923 by Russell. A Polish logician Łukasiewicz first formulated multi-valued logic in 1930. These research studies were carried out long before the assumptions of fuzzy logic which Lofti A. Zadeh originally defined in 1965 (Zadeh 1965), but thanks to his work multi-valued logic was once more discovered. Later, numerous scientists such as Sugeno (1985), Kosko (1997), Kacprzyk and Feddrizzi (1992), Yager (1992), Yamakawa (1989) and others (Bosc and Kacprzyk 1995; Bose 1994; Dubois and Prade 1999; Dubois et al 2002; Larsen 1980; Mamdani 1977; Mendel 1995; Takagi and Sugeno 1998; Zadeh 1999a; Zadeh and Kacprzyk 1992; Zemankowa and Kacprzyk 1993) worked on the idea and further developed it. Also lately, many research works appeared on the use of fuzzy sets, fuzzy logic, and possibility theory for dealing with imprecise information in database management systems (Fuller 1999; Kuncheva and Kacprzyk 2000; Szczepaniak et al 2003; Yu and Kacprzyk 2003). Both theoretical aspects and implemented systems are discussed within the scope of these studies. Since fuzzy logic theory and its applications are covered extensively in literature, only the main features of this theory will be pointed out here.

Fuzzy set theory results from the need to describe complex phenomena or phenomena that are difficult to define and determine using a conventional mathematical apparatus. It is said that it enables to model complex systems using a higher level of abstraction that originates from human knowledge and experience. Traditional reasoning systems based on classical binary logic utilize the *modus ponens* reasoning rule. This rule can be presented as follows:

$$\frac{(P \Rightarrow W), P}{W} \text{ or } \frac{(x \textbf{ is } \mathbf{A} \Rightarrow y \textbf{ is } \mathbf{B}), x \textbf{ is } \mathbf{A}}{y \textbf{ is } \mathbf{B}} \tag{5.1}$$

Bożena Kostek: *Perception-Based Data Processing in Acoustics*, Studies in Computational Intelligence (SCI) **3**, 315–388 (2005)
www.springerlink.com

which means that if *W* results from premise *P* and *P* is true, then inference *W* is also true. In the classical calculus (*p* and *q*) this implication can be expressed as:

$$(p \wedge (p \to q)) \to q \tag{5.2}$$

which can be interpreted as: if *p* is true and $p \to q$ is true then *q* is true.

Fuzzy logic, being an extension of the classical binary logic, introduces the *generalized modus ponens* (GMP) rule which can be written down as follows:

$$\frac{(x \text{ is } \mathbf{A} \Rightarrow y \text{ is } \mathbf{B}), x \text{ is } \mathbf{A}'}{y \text{ is } \mathbf{B}'}, \tag{5.3}$$

where $\mathbf{A}, \mathbf{A}' \subseteq \mathbf{X}$ and $\mathbf{A}, \mathbf{A}' \subseteq \mathbf{X}$ are fuzzy sets defined in non-empty spaces ***X*** and ***Y***, while *x* and *y* are linguistic variables. This means that the premise is: if *x* is *A* then *y* is *B* fact: *x* is *A* consequence: *y* is *B*.

The fuzzy set $\mathbf{B}'$ from the fuzzy logic inference (5.3) is defined as:

$$\mathbf{B}' = \mathbf{A}' \circ (\mathbf{A} \to \mathbf{B}) \tag{5.4}$$

As the fuzzy implication $\mathbf{A} \to \mathbf{B}$ is equivalent to a certain fuzzy relation, the membership function of set $\mathbf{A} \to \mathbf{B}$ can be determined as a composition of the fact and the fuzzy implication operator as expressed below:

$$\mu_{B'}(y) = \sup_{x \in \mathbf{X}} \{T(\mu_{A'}(x), \mu_{A \to B}(x, y))\} \tag{5.5}$$

where *T* is the t-norm (triangular norm) for the logical operation AND. The fuzzy implication inference based on the compositional rule of inference for approximate reasoning, suggested by Zadeh, uses sup-min composition, however in many practical cases sup-*T* is used. The membership function $\mu_{A \to B}(x, y)$ from the previous relation can be expressed on the basis of two known functions $\mu_A(x)$ and $\mu_B(x)$ by one of the following two implications:

- a minimum-type rule, called the Mamdani rule (Mamdani 1977):

$$\mu_{A \to B}(x, y) = \mu_R(x, y) = \min[\mu_A(x), \mu_B(y)] = \mu_A(x) \wedge \mu_B(y) \tag{5.6}$$

- a product-type rule, called the Larsen rule (Larsen 1980):

$$\mu_{A \to B}(x, y) = \mu_R(x, y) = \mu_A(x) \cdot \mu_B(y) = \mu_A(x) \wedge \mu_B(y) \tag{5.7}$$

Other important classes of fuzzy implication operators are: *S*-, and *R*-implications, where *S* is a t-conorm, and *R* is obtained by the residuation of

continuous t-norm *T*. Typical *S*-implications are Lukasiewicz and Kleene-Dienes implications, on the other hand, examples of *R*-implications are Gödel and Gaines implications (Fuller 1999).

Suppose that $X = \{x\}$ is a *universe of discourse*, i.e. the set of all possible elements with respect to a fuzzy concept. Then a *fuzzy subset A* in *X* is a set of ordered pairs $\{(x, \mu_A(x))\}$, where $\{x\} \in X$ and $\mu_A : X \to [0,1]$ is the *membership function* of *A*; $\mu_A(x) \in [0,1]$ is the *grade of membership* of *x* in *A*. A fuzzy variable has values which are expressed in natural language, and its value is defined by a membership function. Since the basic properties of Boolean theory are also valid in fuzzy set theory, they will only be cited here briefly (Kacprzyk and Feddrizzi 1992).

The union of two fuzzy sets *A* and *B* of a universe of discourse *X*, denoted as $A \cup B$ is defined as:

$$\mu_{A \cup B}(x) = \mu_A(x) \vee \mu_B(x), \forall x \in X \tag{5.8}$$

The intersection of two fuzzy sets *A* and *B* of a universe of discourse *X*, denoted as $A \cap B$, is defined as:

$$\mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x), \forall x \in X \tag{5.9}$$

The complement of a fuzzy set *A* of a universe of discourse *X*, denoted as $\neg A$, is defined as:

$$\mu_{\neg A}(x) = 1 - \mu_A(x), \forall x \in X \tag{5.10}$$

The above operations are illustrated in Fig. 5.1. As may be seen in Fig. 5.1, the fuzzy-set intersection is defined as the minimum of the fuzzy set pairs (the smaller of the two elements), the union is defined as the maximum, and the complement produces a reversal in order (Kosko 1997).

Another important notion of fuzzy sets is the size or cardinality of a set *A*. It is defined as:

$$card\ A = \sum_{i=1}^{n} \mu_A(x_i) \tag{5.11}$$

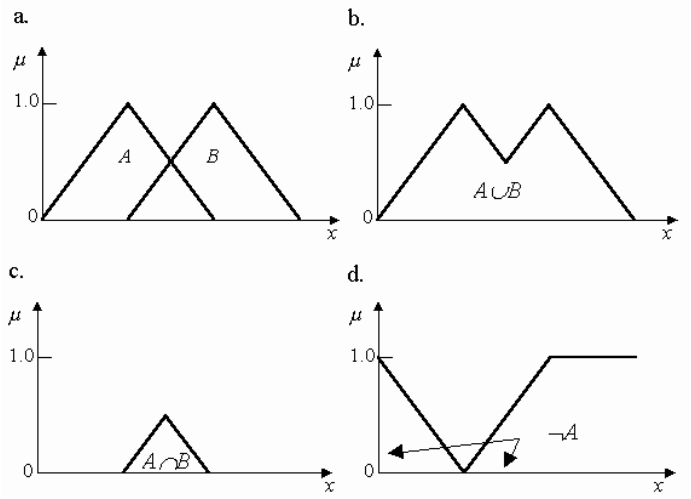


Fig. 5.1. Basic operations in fuzzy theory: Fuzzy sets A and B (a), $A \cup B$ (b), $A \cap B$ (c), $\neg A$ (d)

5.1.1 Fuzzy Reasoning in Control

The primary factor making fuzzy logic predestined for applications in the field of control is the possibility for intuitive modeling of linear and non-linear control functions of optional complication. This capability approximates the decision making process of a machine to that of a human. Fuzzy-based systems also make the description of functions with the use of conditional rules possible.

Typical scheme of data processing based on fuzzy reasoning is presented in Fig. 5.2. The following items describe individual blocks of the fuzzy reasoning system (Takagi and Sugeno 1985).

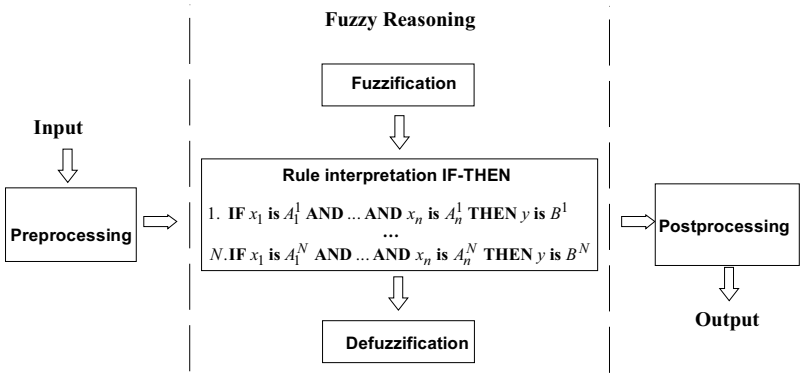


Fig. 5.2. Scheme of fuzzy reasoning system

The literature concerning fuzzy logic is now well developed, so only a short introduction to the principles of fuzzy logic control will be provided here.

Rule Base

The design of fuzzy controllers includes the collection of control rules. These rules consist of linguistic statements which link the controller inputs with their respective outputs. The rules are contained in a rule base. If several linguistic variables are involved in the antecedents and there is one variable in conclusion then the system will be referred to as a multi-input-one-output fuzzy system. The rule base, also called the linguistic model, is a set of N fuzzy rules $\mathbf{R}^{(k)}, k = 1,2,...,N$ of the IF...AND...THEN type, e.g.:

$$\mathbf{R}^{(k)}: \textbf{IF } x_1 \textbf{ is } \mathbf{A}_1^k \textbf{AND} x_2 \textbf{ is } \mathbf{A}_2^k \textbf{AND}...\textbf{AND} x_n \textbf{ is } \mathbf{A}_n^k \textbf{ THEN } y \textbf{ is } \mathbf{B}^k \quad (5.12)$$

where $x_1 \in \mathbf{X}_1, x_2 \in \mathbf{X}_2, ..., x_n \in \mathbf{X}_n$ denote input linguistic variables of the rule basis, $y \in \mathbf{Y}$ is the output fuzzy variable, $\mathbf{A}_i^k$, $\mathbf{B}^k$ are fuzzy subsets in the universe of discourses $\mathbf{X}$, and $\mathbf{Y}$ respectively, for which $\mathbf{A}_i^k \subseteq \mathbf{X}_i \subset \mathbf{R}$ and $\mathbf{B}^k \subseteq \mathbf{Y} \subset \mathbf{R}$, and $\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_n$ and $\mathbf{Y}$ are input and output variable spaces, respectively, while $\mathbf{R}$ denotes the real number set.

A slightly different form of the rule $\mathbf{R}^{(k)}$ was proposed by Takagi and Sugeno. In their rule the antecedent is fuzzy in character, while the functional relation $y_k = f^{(k)}(x_1, x_2, ..., x_n)$ appears as conclusion. This leads to the rule of the following shape (Takagi and Sugeno 1985):

$$\mathbf{R}^{(k)}: \textbf{IF } x_1 \textbf{ is } \mathbf{A}_1^k \textbf{ AND}...\textbf{AND } x_n \textbf{ is } \mathbf{A}_n^k \textbf{ THEN } y_k = f^{(k)}\left(x_1, x_2, ..., x_n\right) \quad (5.13)$$

Discussion in the following part of the chapter will apply to the rules consistent with definition (5.12) unless indicated otherwise. Using designations:

$$\mathbf{x} = \left[x_1, x_2, ..., x_n\right]^T, \ \mathbf{X} = \mathbf{X}_1 \times \mathbf{X}_2 \times ... \times \mathbf{X}_n, \ \mathbf{A}^k = \mathbf{A}_1^k \times \mathbf{A}_2^k \times ... \times \mathbf{A}_n^k \quad (5.14)$$

one can present the *k*th rule $\mathbf{R}^{(k)}$ (5.12) as a fuzzy implication:

$$\mathbf{R}^{(k)}: \mathbf{A}^k \rightarrow \mathbf{B}^k \quad (5.15)$$

The above relation can be interpreted as a fuzzy relation, and therefore the rule $\mathbf{R}^{(k)} \subseteq \mathbf{X} \times \mathbf{Y}$ is a fuzzy set of a membership function:

$$\mu_{R^{(k)}}(\mathbf{x}, y) = \mu_{A^k \rightarrow B^k}(\mathbf{x}, y) \quad (5.16)$$

which can be determined using either the Mamdani (5.6) or the Larsen rule (5.7).

For the given rule base of a control system, the fuzzy controller determines the rules to be fired for the specific input signal condition and then computes the effective control action. Applying inference operators *sup-min* or *sup-prod* (i.e. *supreme-minimum*, *supreme-product*) to the composition operation results in the generation of the control output (Bose 1994).

Pre- and Postprocessing Block

Preprocessing is aimed at converting data fed onto the system input to a format accepted by the reasoning module. Similarly, postprocessing converts data produced by this module to the form consistent with external requirements. The reasoning module itself awaits a sequence of real numbers on input and returns a sequence of real numbers.

Fuzzification Block

Fuzzification is another notion defined in the terminology of fuzzy sets. It can be performed by considering the crisp input values as 'singletons' (fuzzy sets that have a membership value of 1 for a given input value and 0 at other points) and taking the values of the set membership function at the respective data value (Bose 1994). The fuzzification procedure involves transforming the values $\hat{\mathbf{x}} = \left[\hat{x}_1, \hat{x}_2, ..., \hat{x}_n\right]^T \in \mathbf{X}$ of the input signal from the domain of real numbers to the domain of fuzzy sets. To achieve this, one

determines the values of membership functions for subsequent linguistic variables as well as for the given real input value. As a result of the transformation the input value $\hat{\mathbf{x}}$ is mapped into a fuzzy set $\mathbf{A}' \subseteq \mathbf{X} = \mathbf{X}_1 \times \mathbf{X}_2 \times ... \times \mathbf{X}_n$.

Fuzzy Rule Interpretation Block

After the fuzzification phase the fuzzy set $\mathbf{A}' = \mathbf{A}'_1 \times \mathbf{A}'_2 \times ... \times \mathbf{A}'_n$, where $\mathbf{A}' \subseteq \mathbf{X} = \mathbf{X}_1 \times \mathbf{X}_2 \times ... \times \mathbf{X}_n$, is fed onto the input of the fuzzy rule interpretation block. In turn, it delivers to its output N fuzzy sets $\mathbf{B}'^k \subseteq \mathbf{Y}$ equivalent to one resulting fuzzy set $\mathbf{B}' = \mathbf{B}'^1 \cup ... \cup \mathbf{B}'^N$ being the logical sum of N sets $\mathbf{B}'^k$. Reasoning is based on the generalized *modus ponens* rule (5.3), which for the kth rule $\mathbf{R}^{(k)}$ and input signal $\mathbf{x} = [x_1, x_2, ..., x_n]^T$ takes the form:

$$\frac{\left(\mathbf{R}^{(k)} : \mathbf{x} \text{ is } \mathbf{A}^k \Rightarrow y \text{ is } \mathbf{B}^k\right), \mathbf{x} \text{ is } \mathbf{A}'}{y \text{ is } \mathbf{B}'^k} \tag{5.17}$$

Fuzzy set $\mathbf{B}'^k$ is defined by a composition of fuzzy set $\mathbf{A}'$ and relation $\mathbf{R}^{(k)}$, which can be expressed as follows:

$$\mathbf{B}'^k = \mathbf{A}' \circ \left(\mathbf{A}^k \rightarrow \mathbf{B}^k\right) \tag{5.18}$$

Using Eq. (5.5) one can calculate the membership function $\mu_{B'^k}(y)$ of set $\mathbf{B}'^k$ according to the following formula:

$$\mu_{B'^k}(y) = \sup_{\mathbf{x} \in \mathbf{X}} \left\{ T\left(\mu_{A'}(\mathbf{x}), \mu_{A^k \rightarrow B^k}(\mathbf{x}, y)\right) \right\} \tag{5.19}$$

Setting out the resulting fuzzy set $\mathbf{B}'^k$ for the given rule, and therefore calculating the membership function $\mu_{B'^k}(y)$, can be identified with calculating the rule strength. In practical implementations, if the value of the rule strength is zero, the given rule is not fired and is ignored. After the overall rule strength has been calculated, the fired rules are aggregated, which is based on summing the resulting fuzzy sets for all these rules. Fuzzy set $\mathbf{B}'$ obtained in this way is the result set of fuzzy reasoning.

Defuzzification Block

As fuzzy reasoning produces a fuzzy set $\mathbf{B}'$ (or N fuzzy sets $\mathbf{B}'^k$), it needs to be mapped into a single real value $y_o \in \mathbf{Y}$. The *defuzzification* procedure, is a reverse of the fuzzification procedure. It involves the transformation of the values from the domain of fuzzy sets to the domain of real numbers. The operation of defuzzification can be performed by a number of methods of which the center of gravity (centroid) and height methods are most common. The centroid defuzzification method, determines the output crisp value U_0 from the center of gravity of the output membership function weighted by its height $\mu(U)$ (*degree of membership*) and may be described by the following expression:

$$U_0 = \frac{\int U \cdot \mu(U) dU}{\int \mu(U) dU} \tag{5.20}$$

If the Takagi-Sugeno model is used and the rules have the forms presented in definition (5.13), the output value y_o is determined as a normalized weighted average of values of successors $\hat{y}_k$ of subsequent rules, which can be expressed as follows:

$$y_o = \sum_{k=1}^{N} w^k \cdot \hat{y}_k \;/\; \sum_{k=1}^{N} w^k \tag{5.21}$$

where the value $\hat{y}_k$ is calculated for the input signal $\hat{\mathbf{x}} = [\hat{x}_1, \hat{x}_2, ..., \hat{x}_n]^T \in \mathbf{X}$ in the following way:

$$\hat{y}_k = f^{(k)}(\hat{x}_1, \hat{x}_2, ..., \hat{x}_n) \tag{5.22}$$

while the weighing factor w^k depends on the chosen t-norm:

$$w^k = T\left(\mu_{A_1^k}(\hat{x}_1), \mu_{A_2^k}(\hat{x}_2), ..., \mu_{A_n^k}(\hat{x}_n)\right) \tag{5.23}$$

and usually has the following shape:

$$w^k = \min\left(\mu_{A_1^k}(\hat{x}_1), ..., \mu_{A_n^k}(\hat{x}_n)\right) \text{ or } w^k = \mu_{A_1^k}(\hat{x}_1) \cdot ... \cdot \mu_{A_n^k}(\hat{x}_n) \tag{5.24}$$

Designing Fuzzy Systems

Individual steps of designing fuzzy systems are presented below. It is worth noting that some of the tasks listed below are performed empirically. In spite of the simple and natural structure of fuzzy systems, methods to choose a membership function of optimal shape and rule base remain unknown. The idea of fuzzy logic basically comes down to replacing the output function descriptions for all possible input states with a group of membership functions which represent certain ranges or sets of input values.

The process of creating a fuzzy logic application is usually comprised of six stages:

- formulating the problem and identifying control signals which define the system behavior,
- defining linguistic variables and the corresponding fuzzy attributes
- defining the inference rules,
- designing the membership function for each variable,
- constructing the rule base and rule processing,
- computing the values of control signals in the defuzzification process.

The membership functions are standard and may be defined by stating three parameters:

- Center Location - *central value*,
- Width,
- Type: *inclusive/exclusive*.

The meaning of these parameters is illustrated in Figs. 5.3 and 5.4.

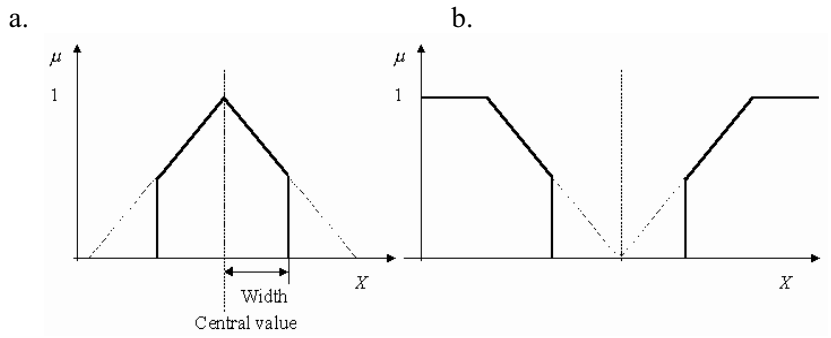


Fig. 5.3. Shape of the membership function: inclusive type (a), exclusive type (b)

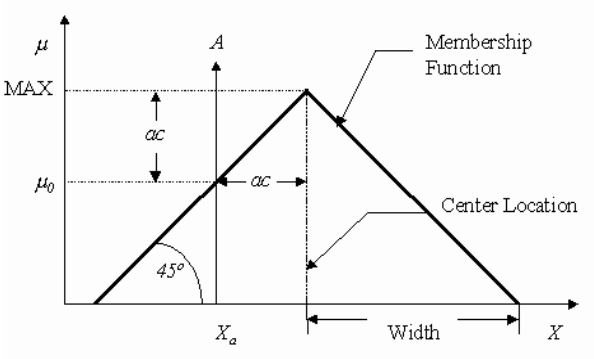


Fig. 5.4. Membership function parameters

As is depicted in Fig. 5.4, the membership function may have a triangular shape which simplifies the process of computing its value. The degree of membership is in this case a simple function of distance *ac* from the input value to central value X_a (see Fig. 5.4). Distance *ac* is then subtracted from the maximum value of the membership function *MAX*. Hence the membership degree amounts to:

- for a function of the inclusive type:

  $\mu = MAX - abs(ac)$; when $abs(ac) \leq width$

  $\mu = 0$; when $abs(ac) > width$

- for a function of the exclusive type:

  $\mu = MAX$; when $abs(ac) > \mu_0$

  $\mu = MAX - \mu_0 + abs(ac)$; when $\mu_0 \geq abs(ac) \geq width$

  $\mu = 0$; when $abs(ac) < width$

Fuzzy processing is based on a set of inference rules, and there are several ways to create sets of these rules. Most frequently, they are created heuristically rather than by using closed mathematical formulas, which is why this process is difficult to automate. Nonetheless, three directions can be formulated:

- representation of human knowledge and experience,
- usage of analytical bases,
- formulation of generalizations.

The inference process, based on fuzzy logic rules, may be illustrated as follows (Hua and Yuandong 1994).

Let x_1 and x_2 be input variables, and y the output variable;

Rule 1: IF x_1 belongs to A_{11} AND x_2 belongs to A_{12}, THEN y belongs to B_1

Rule 2: IF x_1 belongs to A_{21} AND x_2 to A_{22} THEN y belongs to B_2

The values of these particular rules are defined in the formulas:

$$w_1 = \min(\mu_{A_{11}}(x_1), \mu_{A_{12}}(x_2)) \tag{5.25}$$

$$w_2 = \min(\mu_{A_{21}}(x_1), \mu_{A_{22}}(x_2)) \tag{5.26}$$

A graphic illustration of the inference process is depicted in Fig. 5.5.

The actual output value that results from the completed inference is computed as:

$$y = \frac{\sum_{i=1}^{2} w_i y_i}{\sum_{i=1}^{2} w_i} \tag{5.27}$$

where:

$$y_1 = \mu_{B_{1-1}}(w_1), \; y_2 = \mu_{B_{2-1}}(w_2) \tag{5.28}$$

A graphic illustration of the defuzzification process is depicted in Fig. 5.6.

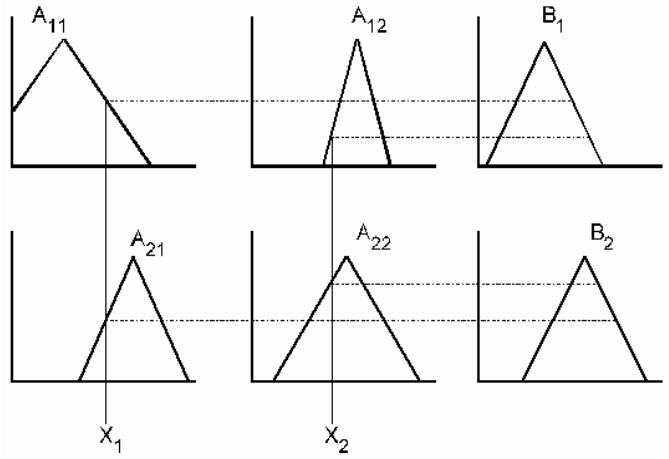


Fig. 5.5. Graphic illustration of fuzzy logic rule-based operations

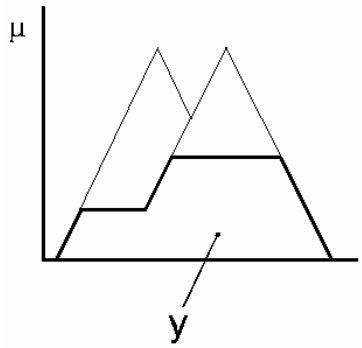


Fig. 5.6. Graphic illustration of the defuzzification phase. Computation of output value y is based on the sets resulting from the fulfillment of the rules

In some applications, a hybrid method comprised of both fuzzy and mathematical approaches may be used. As an example of such a method, the relational method introduced by Sugeno (1985) may be cited. The principles of this method are shown in Fig. 5.7. There are two inputs in the engineered system, namely: Width (W) and Height (H). The output (I_s) is, in this case, a combination of rule sets and linear equations, because it is assumed that there are some regions in which the outputs may be expressed as linear functions of the inputs. Consequently, the *IF* part of the rule comprises a fuzzy expression, but the *THEN* portion is a linear combination of inputs and constant coefficients, the latter derived from analysis and tuned by observation. Rules 1 and 2 in Fig. 5.7 are as follows:

RULE1: IF W is *MEDIUM* AND H is *MEDIUM* THEN $I_{s1}=A_{01}+A_{11}W+A_{21}H$

RULE2: IF W is *ZERO* AND H is *MEDIUM* THEN $I_{s2}=A_{02}+A_{12}W+A_{22}H$

The last task to be performed in order to determine the precise output is the defuzzification process, which in this case is a weighted average of linear equations. It is given that the relational method requires fewer rules and gives better accuracy than the rule base method.

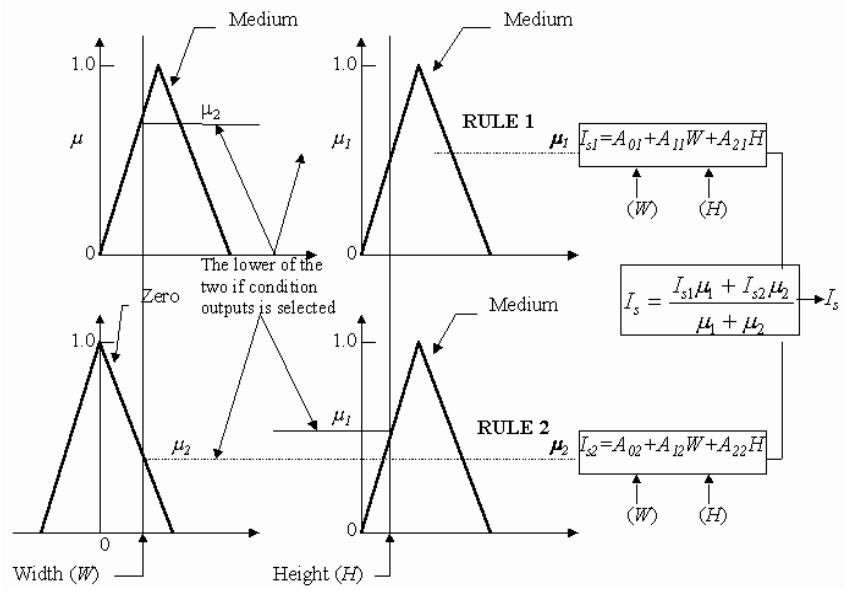


Fig. 5.7. Relational method illustration (Bose 1994)

Some of those theoretical assumptions will be shortly illustrated by an example of the author's work taken from the research done previously (Kostek 1999).

5.1.2 Fuzzy Control of Pipe Organ

Since one of the unique control applications of fuzzy logic techniques to musical acoustics is the fuzzy control of a classical pipe organ (Kostek 1994b), it is perhaps worth mentioning. Computerizing classical pipe organs opens a new domain of interests, in which modern technology meets the traditional way of playing such instruments. The application of a microprocessor system to an organ may significantly improve many of the control and display functions of the console. Such a computer control was engineered by the author for the classical pipe organ instrument from a Musical Theater in the south of Poland.

Computer control enables a new approach to the problem of limitations existing in musical articulation of a pipe instrument with an electromagnetic action. This kind of a pipe organ control is characterized by the promptness in the pipe response, as the air flow cannot be controlled otherwise than by rapidly opening and closing the valve. In the opinion of or-

ganists, this deprives them of the possibility to interpret music according to their wishes.

The process of a pipe organ activation, which consists of depressing a key, sound rising in a pipe and the reaction of a valve, is difficult to describe mathematically (Caddy and Pollard 1957; Kostek 1992, 1994a, 1994b, 1997; Kostek and Czyzewski 1991, 1992, 1993). In addition, since these processes are imprecise in nature, a typical microprocessor-based organ control system may therefore be replaced by a learning control system capable of modeling the non-linearities gained from entries defined and related decisions. Consequently, fuzzy logic techniques may be employed in a pipe organ control system. Such a system was engineered and applied to a pipe organ model within the research work done by the author in 1993-1994 under the support of the Committee for Scientific Research, Warsaw, Poland (Kostek 1994b).

For the purpose of this study, a model of a pipe organ was designed and constructed (Kostek 1997, 1999). It consists of two elements: a model of an organ tracker action and a control system based on a fuzzy logic technique (Fig. 5.8). The model of the organ is made from oak, and consists of: bellows with a volume of 0.06m^3, covered with leather (the bellows are filled with air through a foot pedal); a wind chest sized 0.4m x 0.3m x 0.2m; two organ pipes (Principal 8' - tin pipe, and Bourdon 8' - wooden pipe); and a tracker action which enables both mechanical control and electrical activation. Three electromagnets used in this control system are combined electrically to one key. The valve is driven by electromagnets with a counteracting spring. Electric activation is obtained through the use of a set of electromagnets controlled by a system constructed on the basis of fuzzy logic. Activating the electromagnets causes the air inflow to a selected pipe. A block diagram of the system which controls the electromagnets of the organ pipe valves is shown in Fig. 5.9. Additionally, the system configuration is shown in Fig. 5.10. The following components are included: a dynamic keyboard sensitive to the velocity of key motion and connected through a MIDI interface to a computer; a PC computer with software controlling the FUZZY microprocessor card; a FUZZY microprocessor card and a MIDI interface card installed in a PC computer; a specially constructed control display of a key number and velocity; a buffer of information exchanged between the MIDI and FUZZY cards; and a buffer to control the electromagnets via the transistor drivers (Fig. 5.10). The applied Yamaha PSR-1500 MIDI keyboard is of a *touch-sensitive* type, therefore according to the velocity with which the key was pressed a MIDI code is generated. A sensor under the keyboard picks up the signal correlated to the way of depressing the key and at the same time transforms it into the system input signal.

Fig. 5.8. Fuzzy logic-based control system for a pipe organ (Kostek, 1999)

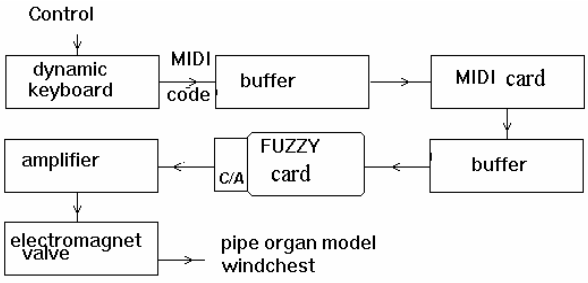


Fig. 5.9. Block diagram of the control system

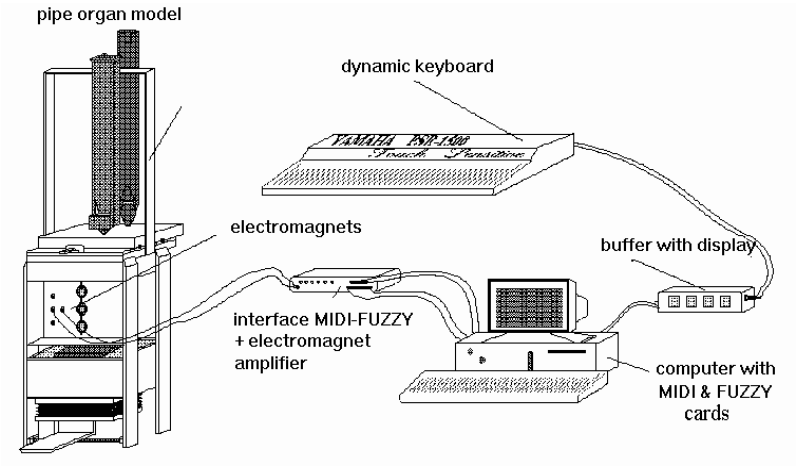


Fig. 5.10. Layout of the fuzzy logic-based control system configuration

The information on pressing or releasing the key is transmitted from the keyboard through the MIDI interface in the form of 2 or 3 bytes of data:

- the first byte – the command meaning that data will be transmitted,
- the second byte – the information on the key number within the range from 0 to 127,
- the third byte – the information on the velocity of pressing the key, in the range from 1 to 127.

The key number is essential because of the relation between the size of the pipe and the articulation artefacts. In traditional, mechanical organs, articulation features appear mostly in low tones. The sound rise in large pipes may be fast or slow, so it is possible to hear the differences in the articulated sounds. Small pipes, because of their size, are excited by the wind blow very quickly and speak nearly always in the same way.

The above information is decoded by the computer through a MIDI decoding procedure. Obtained values are periodically transmitted to the fuzzy logic control system at the speed of 31.25 kBaud. The total transmission time t (Eq. 5.29) consists of at least three delays, namely:

- t_1 is the result of data transmission from the keyboard to the MIDI card:

$$t_1 = \frac{20\text{bit}}{31250\text{bit/s}} = 640\mu\text{s} \tag{5.29}$$

- t_2 corresponds to the data processing in the MIDI card:

$t_2 \approx 30\mu s$

- t_3 is needed for the data processing in the FUZZY microprocessor card:
  $t_3 \approx 8\mu s$

$$t \approx t_1 + t_2 + t_3 \approx 640\mu s + 30\mu s + 8\mu s \approx 678\mu s \tag{5.30}$$

As shown in Fig. 5.10, three parallel connected electromagnets are applied to drive the pallet opening the air inflow. The electromagnets are switched on and driven by the current, the value of which is defined by the fuzzy rule system. Thus, any key motion rates will be translated into the way the valve is being open, and in consequence into the air pressure in the pipe that is decisive to the character of the rising sound.

Two parameters that are extracted periodically from the MIDI code, namely the key number and the velocity, create two fuzzy inputs, labeled as:

INPUTS:
KEY_NUMBER; *VELOCITY*,

and output is associated with the current applied to electromagnet coils and is denoted *CURRENT*. Corresponding membership functions are labeled as follows:

OUTPUT:
LOW_CURRENT; *MEDIUM_CURRENT*; *HIGH_CURRENT*.

The fuzzifiers were named as follows:

FUZZIFIERS:
for *KEY_NUMBER* and *VELOCITY* : - *LOW*
- *MEDIUM*
- *HIGH*

The output of the system is set at the beginning to 0. The MIDI code assigns the keys with numbers from a range starting from 0 (when no key is pressed) to 127. The mapping of the keyboard was reflected as *KEY_NUMBER*, and is presented in Table 5.1. The velocity values are represented as in Table 5.2.

Table 5.1. Keyboard mapping

KEY_NUMBER	CENTER	WIDTH
LOW	30	29
MEDIUM	70	25
HIGH	100	27

Table 5.2. Velocity mapping

VELOCITY	CENTER	WIDTH
LOW	30	29
MEDIUM	70	15
HIGH	101	26

The above listed values (Table 5.1 and 5.2) were set experimentally. The performed experiments allow to show the plot of membership functions corresponding to the inputs *KEY_NUMBER* and *VELOCITY* and to *CURRENT* denoted as OUTPUT (Fig. 5.11). As can be seen from Fig. 5.11, triangular membership functions are employed in the fuzzy controller.

The inputs and fuzzifiers are producing terms that are used in the following rules:

RULES:

if *KEY_NUMBER* **is** *OFF* **then** *0*
if *VELOCITY* **is** *OFF* **then** *0*
if *KEY_NUMBER* **is** *LOW* **and** *VELOCITY* **is** *LOW* **then** *LOW_CURRENT*
if *KEY_NUMBER* **is** *MEDIUM* **and** *VELOCITY* **is** *LOW* **then** *LOW_CURRENT*
if *KEY_NUMBER* **is** *HIGH* **and** *VELOCITY* **is** *LOW* **then** *MEDIUM_CURRENT*
if *KEY_NUMBER* **is** *LOW* **and** *VELOCITY* **is** *MEDIUM* **then** *MEDIUM_CURRENT*
if *KEY_NUMBER* **is** *MEDIUM* **and** *VELOCITY* **is** *MEDIUM* **then** *MEDIUM_CURRENT*
if *KEY_NUMBER* **is** *HIGH* **and** *VELOCITY* **is** *MEDIUM* **then** *HIGH_CURRENT*
if *KEY_NUMBER* **is** *LOW* **and** *VELOCITY* **is** *HIGH* **then** *HIGH_CURRENT*
if *KEY_NUMBER* **is** *MEDIUM* **and** *VELOCITY* **is** *HIGH* **then** *HIGH_CURRENT*
if *KEY_NUMBER* **is** *HIGH* **and** *VELOCITY* **is** *HIGH* **then** *HIGH_CURRENT*

Each rule produces a number which is calculated according to fuzzy logic principles from the cross-section of the input values with the membership functions (see Fig. 5.11). The winning rule is one that has the highest value assigned during the calculations. On the basis of the terms adopted, the numerical values are converted to the respective current which is driving the electromagnets. This means that the lowest output value causes the slowest opening of the valve, while other values appear-

ing on the output, which match other terms, result in opening the valve faster.

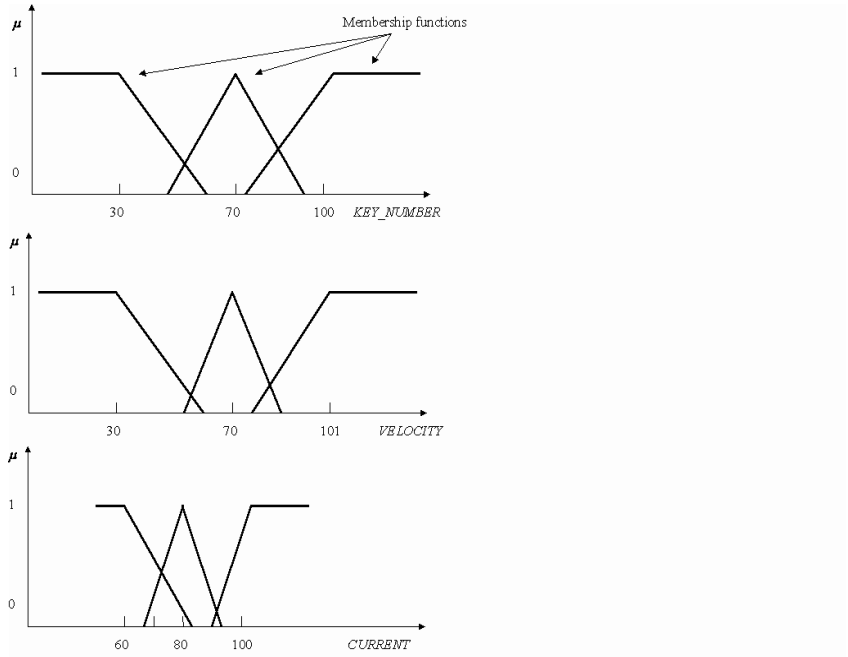


Fig. 5.11. Membership functions corresponding to *VELOCITY* (a), *KEY_NUMBER* (b) inputs and to *CURRENT* denoted as output (c), where: μ - degree of membership

Recordings of the signals generated by the model were made based on the system whose block diagram is presented in Fig. 5.12. A pair of sensors was attached to the key, and activated electrically. The input of the system was controlled through a *touch-sensitive* keyboard. Impulses from the sensors responsible for the time of depressing the key were registered. The value of velocity of depressing the key was read from the MIDI interface display. The output signal from the control system was recorded on the left channel of the tape recorder, while the sound of the pipe was registered on the right channel.

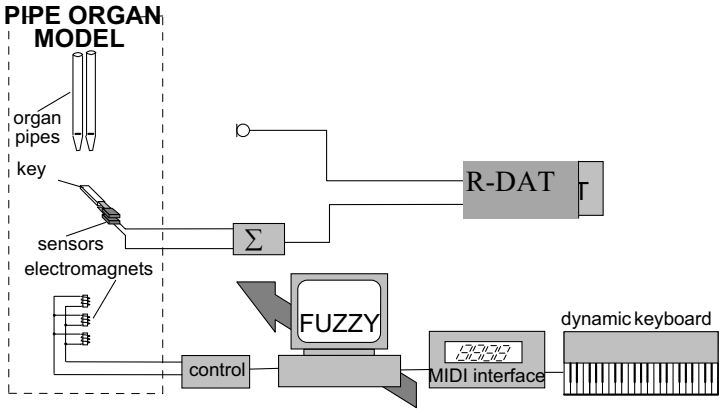
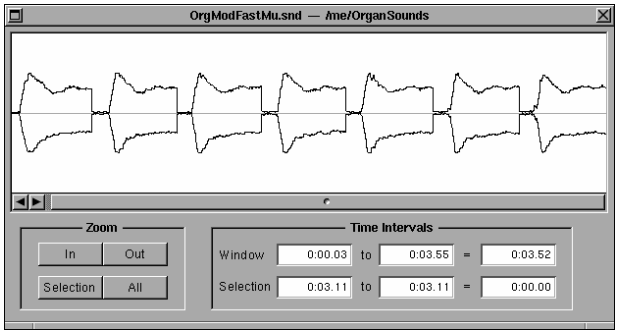


Fig. 5.12. Block diagram of the recording system of the pipe organ model

Examples of analyses of the time- and frequency-domain characteristics of the recorded sounds are presented in Figs. 5.13 and 5.14.

a.



b.

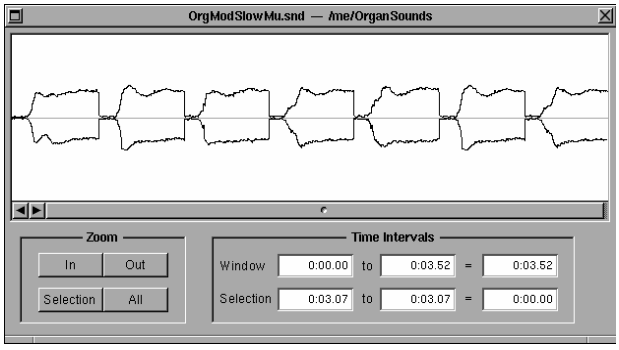
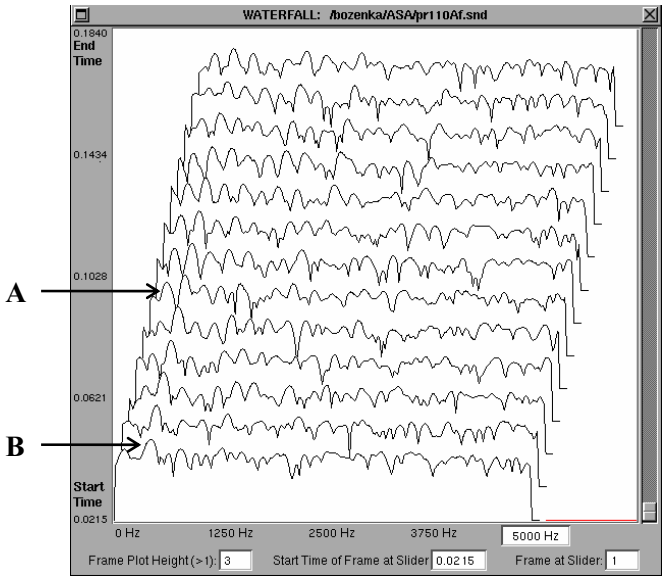


Fig. 5.13. Analyses of time-domain characteristics of the sounds of Principal 8' in the case of: fast opening of the valve (a), slow opening of the valve (b)

The plots show the differences that are visible in the time representation of the analyzed sounds, as well as in the representation of waterfall plots, respectively for fast (Figs. 5.13a and 5.14a) and slow (Figs. 5.13b and 5.14b) opening of the valve.

a.



b.

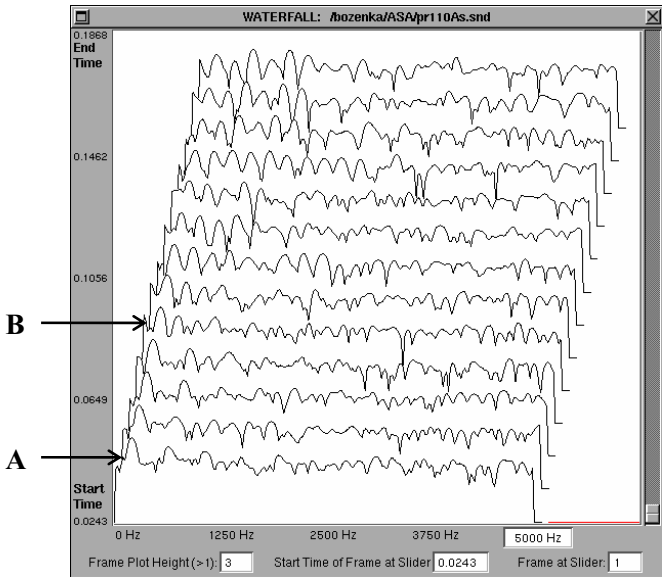


Fig. 5.14. Analyses of frequency-domain characteristics of the sounds of Principal 8' in the case of: fast opening of the valve (a), slow opening of the valve (b)

Both spectral characteristics differ mainly in the behavior of the second harmonic whose dynamic of change depends directly on the rate at which the key is depressed – the faster the key is depressed, the quicker the second harmonic grows. There are also other discrepancies. It is easy to observe that the fundamental is much weaker when the key is depressed quickly. Arrows 'A' in Fig. 5.14 show the starting point of the rising of fundamentals, whereas arrows 'B' show the rising of the second harmonics. Additionally, in Fig. 5.15 the adequate sonogram analyses are illustrated. The difference in starting attacks in fast and slow opening of the valve is clearly visible.

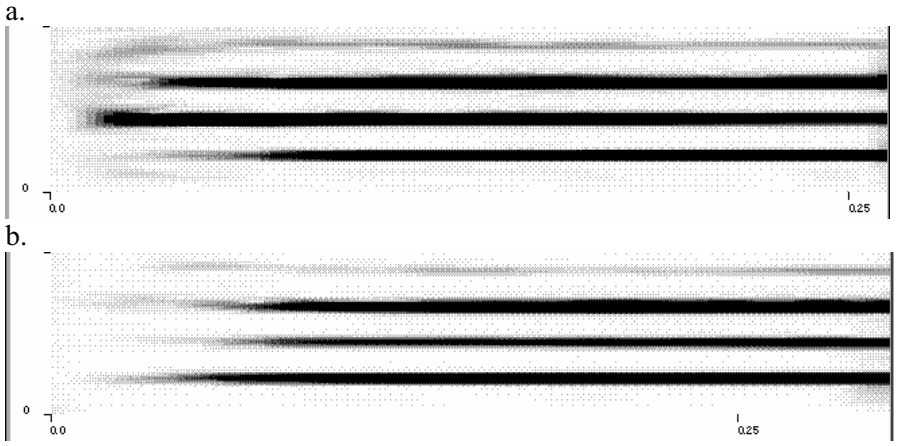


Fig. 5.15. Sonograms of sounds recorded from the model: fast (a) and slow (a) opening of the valve. The horizontal axis represents the time-domain (0-300ms), and the frequency-domain is depicted in vertical axis (0-1000Hz), the intensity of shades depicts the magnitude of particular harmonics (white color corresponds to -60dB)

Therefore, it may be said that the constructed fuzzy logic control system for a pipe organ action responds properly depending on differentiated musical articulation, enriching music by providing nuances to its interpretation.

5.2 Rough-Fuzzy-Based Classifier of Musical Timbre

As mentioned before, 'computing with words', a concept introduced by Zadeh refers to the fact that humans employ words in computing and reasoning, arriving at conclusions formulated in words from premises expressed in a natural language (Zadeh 1994, 1996, 1999b). Computing with

words is generally less precise than computing with numbers, but human perception is also imprecise and imperfect. It seems that this new paradigm of computing can be used with success in musical acoustics as it offers better processing of subjective descriptors of instrument sounds and enables the analysis of data

While assessing the quality of musical instrument sounds, humans use criteria that are rarely quantitative but most often qualitative. Therefore, there is a need to find methods that make it possible to process such descriptive notions as bright, dark, clear, soft, high, low, etc., with techniques that are adequate for this task (Kostek 2003). Soft computing offers techniques that have been developed and tested in many other domains and applications (Pawlak 1982; Skowron 1994; Zadeh 1996, 1999b; http://www.soft-computing.de/def.html; http://www.pa.info.mie-u.ac.jp/WFSC/).

Relationships between the objectively measured parameters of musical instrument sounds and their subjective quality as assessed by listeners (preferably experts in musical domain) cannot in most cases be crisply defined. This leaves a wide margin of uncertainty which depends on individual preferences and unknown influences of particular parameter values on the timbre quality of the tested sound. However, an attempt has been undertaken to show how to better process qualitative notions in musical acoustics.

5.2.1 Musical Instrument Sound Subjective Descriptors

The notion of multidimensionality of musical sound timbre has been already reviewed in Chapter 2. One can refer to studies by numerous researchers (McAdams and Winsberg 1999; De Bruijn 1978; Cook 1999; Cosi et al 1984; Grey 1977; Iverson and Krumhansl 1993; Jensen 2001; Krimphoff et al 1994; Misdariis et al 1998; De Poli et al 1991, 2001; Pratt and Doak 1976; Pratt and Bowsher 1978; Reuter 1996; Vercoe et al 1998; Wessel 1979). Computer processing can easily deal with multidimensionality of feature vectors containing musical sound descriptors (Iverson and Krumhansl 1993; Jensen 2001; Kostek 1999, 2003; Kostek and Czyzewski 2001a; Lindsay et al 2001). Moreover, such processing is now highly needed for automatic queries within the digital archives of sound. More dimensions can help to distinguish between particular instruments or musical instrument groups. Another vital problem related to discovering the relationship between sound descriptors and objectively derived parameters remains unsolved. Only a few parameters such as for example brightness have their unquestioned interpretation – this subjective descriptor is related

to spectral centroid (Beachaump 1993). Further discussion is also required on the assignment of sound parameter ranges. This will also serve to better distinguish between musical instrument sound characteristics.

5.2.2 Methodology

There are many tests organized that aimed at subjective assessment of sound timbre or sound quality. Two important assumptions were made in tests carried out in the multidimensional perceptual scaling of musical timbre. Sound stimuli were synthesized, and they were equalized for perceived pitch, loudness and duration. This was done in order to get reliable results during timbre scaling tests. However, in the presented study it is assumed that the relationship between subjective descriptors and objectively derived parameters will serve for better quantization of numerical values. In such a case testing should be done using natural sound stimuli. Such tests were carried out in architectural acoustics in order to describe the quality of an interior. It was possible to assign labels to certain numerical parameter values by experts. Such tests were arduous but resulted in reliable evaluation of acoustical parameters. It is much easier for experts to say that such a sound has 'dark' or bright' quality, and contrarily it is difficult to assign numerical values. The problem remains 'how bright' the sound is or what 'nasal' or 'flute-like' quality means as expressed in numbers.

Let us consider how such a procedure could be carried out. First, one should choose such attributes that have subjective meaning to experts. A few such parameters were already found and named in the musical acoustics domain and they are based on parameters derived from time, frequency and time-frequency domains. This can be a starting point to list some parameters suitable for a feature vector both in subjective and objective domain.

Now, the problem is not only to assign ranges to such parameters (using word descriptors) – one can easily imagine that experts would unanimously decide what high pitch of a certain musical instrument means. In such a procedure a subject has to associate presented stimuli with a set of adjective scales (semantic). The subject's task is to indicate for each sound a three- or five-point scale, which of the given terms applies to the stimulus. The drawback is that experts are forced to judge the stimuli in terms of prescribed semantic categories and scales. The preselection of scales determines the resolution of the analysis while the verbal categories may seem different from the expert's auditory sensation. In addition one should be aware that building such a set of parameters could be done only ex-

perimentally. Even an expert in musicology cannot decide as to the number of parameters and their significance to the instrument recognition without subjective tests and processing of results.

Another problem is to find rules on the basis of which a chosen instrument can be qualified into an adequate group with only some degree of uncertainty. For this purpose both computing with words concept and processing using soft computing methods can be applied. A discussion of the main points of such an approach will be shown further on.

5.2.3 Rough-Fuzzy Processing of Test Result Processing

Relationships between the objectively measured parameters of musical instrument sounds and their subjective quality as assessed by listeners (preferably experts) cannot in most cases be crisply defined. This leaves a wide margin of uncertainty which depends on individual preferences and unknown influences of individual parameter values on the timbre quality of the tested sound. Consequently, the results of subjective tests had to be processed statistically (hitherto used approach) in order to find links between preferences and concrete values of parameters representing the objective features of tested objects.

A new extended proposal of a procedure for analyzing subjective test results is formulated. In the first step of the analysis, the results of listening test sessions should be collected into tables, separately for each expert and for each sound excerpts. Then, these tables should be transformed into the format of the decision table used in the rough set decision systems (Table 3.10). Objects t_1 to t_n from Table 3.10 represent now various musical instrument sounds, and attributes A_1 to A_m are denoted as tested sound characteristics. The expert's scoring is defined by the grades a_{11} to a_{nm} (quantized values are labeled descriptively as for example low, medium, and high). The decision D is understood as a value assigned to the name of a musical instrument or a number referring to it. The questionnaire form that can be used in listening tests is as presented in Table 5.3. Subjects are asked to fill in the questionnaire during listening sessions. Having collected the assessments of perceptual dimensions of the tested sounds from experts, it is possible to create a decision table. The result of the rough set-based processing is a set of rules that will be later used to recognize a musical instrument sound unknown to the system (see Fig. 5.16).

Table 5.3. Listening test results for a given sound No. *i* (Denotations: S/G/D - Subject/Grades/Descriptors P - *Pitch*, Br. - *Brightness*, At. - Attack *asymmetry*, T/NQ - *Tone*/*noise-like-quality*, At.d. – *attack duration*, V - *Vibrato*, S - *Synchronicity*, *Inh.* - *Inharmonicity*, M.Instr.Cl. - *Musical Instrument Class*)

S/G/D	P	Br.	At.	T/NQ	....	At.d.	V	S	Inh.	M.Instr. Cl.
1	low	low	low	low	....	low	low	high	low	No. 1
i	....	....	....	....	....	....	....	....	....	....
n	med.	high	high	low	.....	med.	high	high	high	No. 4

It is worth observing the distribution of parameter values of various musical instruments, for example in Fig. 5.17 a distribution of attack duration for the staccato articulation for 11 musical instruments is shown.

It is obvious that with the increase of pitch, there is a change of subjective perception of timbral characteristics, therefore it is important to evaluate particular descriptors as functions of pitch. It is probably that during such tests some additional requirements as to the testing procedure of descriptors should be defined. In Table 5.4 an example of such a division of classes to be tested with regard to pitch is shown. In addition, in Table 5.5 classes of pitch are assigned to analyzed instruments.

The decision table should be processed using the rough set method. In this way, a set of rules would be created, which may subsequently be verified by experts.

Table 5.4. Division of pitch into classes

Class No.	Pitch range		Class No.	Pitch range	
	from:	up to:		from:	up to:
1	C2	D2	12	A4	B4
2	D#2	F2	13	C5	D5
3	F#2	G#2	14	D#5	F5
4	A2	B2	15	F#5	G#5
5	C3	D3	16	A5	B5
6	D#3	F3	17	C6	D6
7	F#3	G#3	18	D#6	F6
8	A3	B3	19	F#6	G#6
9	C4	D4	20	A6	B6
10	D#4	F4	21	C7	D7
11	F#4	G#4	22	D#7	G7

Table 5.5. Classes of pitch assignment for analyzed instruments

Instrument	Pitch range	Classes of pitch
Oboe	D3 – G6	8 – 19
Cello	C2 – G#5	1 – 15
Alto	C3 – D7	5 – 21
Violin	G3 – G7	7 – 22
English horn	E3 – A5	6 – 15
French horn	D2 – E5	1 – 13
Saxophone	C#3 - A5	5 – 15
Clarinet	D3 – F6	5 – 18
Bassoon	A2 – D5	4 – 13
Trombone	E2 – E5	2 – 14
Trumpet	E3 – G#5	6 – 15

A decision system based on rough set theory engineered at Gdansk University of Technology can be used for this purpose (Czyzewski 1998, 2002). It includes learning and testing algorithms. During the first phase, rules are derived to become the basis for the second phase. The generation of decision rules starts from the rules of length 1, continuing with the generation of rules of length 2, etc. The maximum rule length may be determined by the user. The system induces both possible and certain rules. It is assumed that the rough set measure Eq. (5.31) for possible rules should exceed the value 0.5. Moreover, only such rules that are preceded by some shorter rule operating on the same parameters are considered.

A rough set measure of the rule describing concept X is the ratio of the number of all examples from concept X correctly described by the rule:

$$\mu_{rs} = \frac{|X \cap Y|}{|Y|} \tag{5.31}$$

where X is the concept, and Y denotes a set of examples described by the rule.

In the testing phase the leave-one-out procedure is performed. During the jth experiment, the jth object is removed from every class contained in the database, the learning procedure is performed on the remaining objects, and the result of the classification of the omitted objects by the produced rules is saved.

In the rough set-based processing discretized data is used. This means a process of replacing the original values of input data with the number of an interval to which a selected parameter value belongs. These methods have been presented in Chapter 3. One can also refer to studies by Cosi et al. (1994) in which they use Self Organizing Maps (SOM) for timbral data mapping. However, in the proposed method data is quantized by means of

labels assigned by experts in listening tests, so there is no need to discretize them at this stage of analysis. The mapping process of test results to fuzzy membership functions will be presented later on but is seen on the right side of Fig. 5.16a. The second phase of the expert system (Fig. 5.16b), namely automatic classification of musical instrument will also be explained further on.

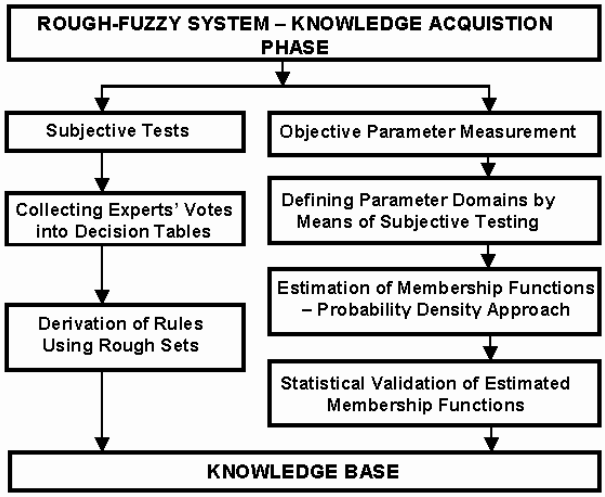
In Fig. 5.17 the distribution of one of the parameters gathered in the system for 11 instrument classes is shown.

The rules are of a form:

RULES:

if (*Pitch* = high) & (*Brightness* = high) & ..... then (*Class* = No. 1)
μ_{rs}=0.9
if (*Pitch* = med) & (*Brightness* = high) & ..... then (*Class* = No. 1)
μ_{rs}=0.8
.....
if .... & (*Synchronicity* = high) & (*Inharmonicity* = high) then (*Class*= No. 4)
μ_{rs}=0.9

a.



(Legend to Fig. 5.16, see next page)

b.

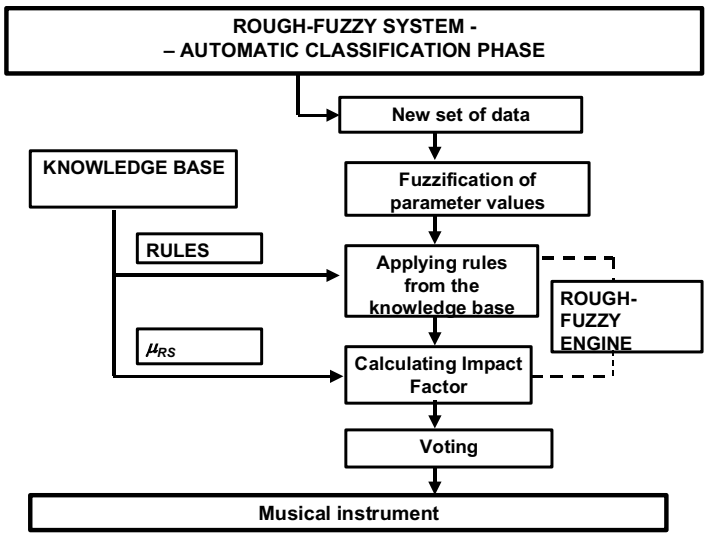


Fig. 5.16. Rough-fuzzy expert system: knowledge acquisition phase (a), automatic classification phase (b)

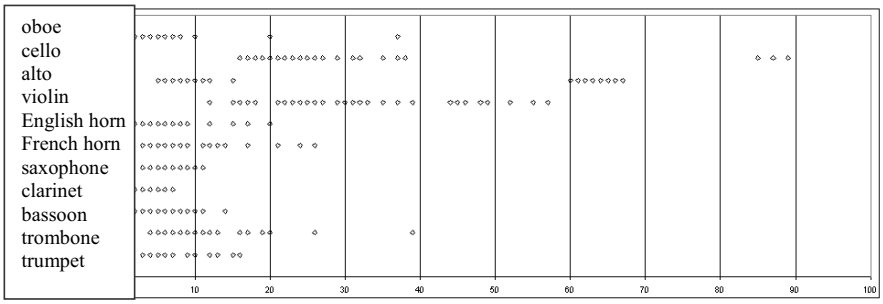


Fig. 5.17. Distribution of attack duration values for 11 musical instrument classes for staccato articulation

Mapping Test Results to Fuzzy Membership Functions

The next step of the procedure is to obtain subjective ratings for each of objective parameters as assessed separately from others. The mapping of objective parameter values to their subjective assessments by many experts creates some fuzzy dependencies, which can be represented by fuzzy membership functions corresponding to each parameter separately.

As was mentioned before, experts, while listening to a sound, are instructed to rate their judgements using such descriptions as low, medium, and high, etc. The procedure uses a concept of the Fuzzy Quantization Method (FQM) applied to acoustical parameters (Kostek 1999). This results in the relation of semantic descriptors to the particular parameter quantities. The distribution of the observed instances very often suggests the trapezoidal or triangular shape of a membership function (see sample membership functions presented in Fig. 5.18).

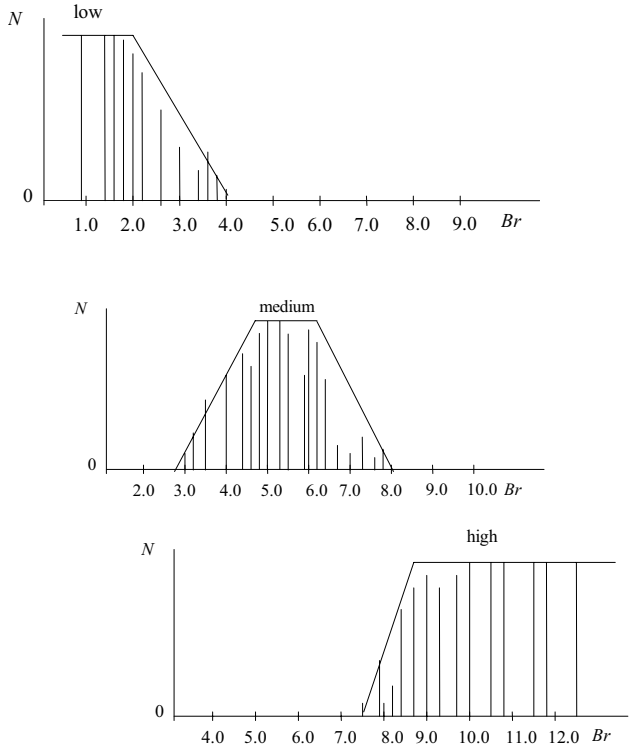


Fig. 5.18. Experts' vote for the *Brightness* parameter, *N* - number of experts voting for particular values of *Brightness* (*Br*)

One of the important tasks is to approximate the tested parameter distribution. This can be done by several techniques. The most common technique is a linear approximation, where the original data range is transformed to the interval of [0,1]. Thus, triangular or trapezoidal membership functions may be used in this case. Also, polynomial approximation is often used for such a purpose. Another approach to defining the shape of the

membership function involves the use of the probability density function. The last mentioned technique was very thoroughly discussed in the author's previous work (Kostek 1999).

Automatic Classification Phase

In order to enable the automatic recognition of a musical instrument class, new data representing sound parameter values is fed to the system inputs (Table 5.6). The first step is the fuzzification process in which degrees of membership are assigned for each crisp input value (as in Fig. 5.18). Therefore, for the data presented in Table 5.6, the degree of membership for each input value (for a given label) has to be determined (Kostek 2003).

Table 5.6. Set of parameter values presented to the system inputs (Denotations: No. - Sound No. *i*, P- *Pitch*, Br. - *Brightness*, At. - *Attack asymmetry*, T/NQ - *Tone*/*noise-like-quality*, V- *Vibrato*, S - *Synchronicity*, Inh. - *Inharmonicity*)

No.	P	Br.	At.	T/NQ	...	V	S	Inh.
1	440	7.75	.....	.....	.....	.....	.....	.....

The pointers in Fig. 5.19 refer to the degrees of membership for the precise value of *Brightness*=7.75. Thus, when the value of *Brightness* equals 7.75, it belongs, respectively, to the low fuzzy set with the degree of 0, to the medium fuzzy set with the degree of 0.4 and to the high fuzzy set with the degree of 0.6. The same procedure should be applied to other parameters of Table 5.3.

It should be remembered that after the rough set processing, only the strongest rules with a rough set measure value exceeding 0.5, would be considered. The rough set measure associated with the individual rule will be contained in the knowledge base. Since there might be several rules corresponding to the given musical instrument, thus during the classification phase, a so-called Impact Factor will be calculated. It is equal to the sum of rough set measures of rules associated with an individual musical instrument. This means that even if a rule is certain for a given instrument (in this case rough set measure equals 1), but there exist several rules that would point out another instrument, after the aggregation of rough set measures (Impact Factor), the latter instrument name would be returned in the system decision. This procedure refers to 'Voting' in the block-diagram shown in Fig. 5.16b.

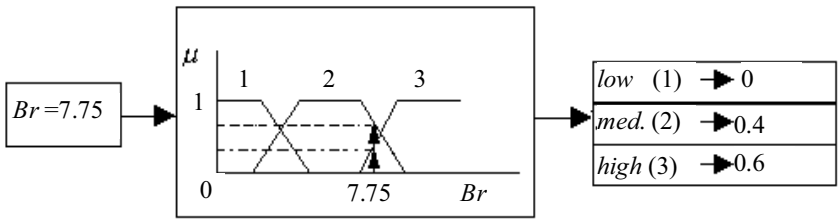


Fig. 5.19. Fuzzification process of the *Brightness* parameter

With rules derived from the rough set decision table and membership functions determined empirically for the studied parameters, one can create an expert system that provides automatic decisions on musical instrument classification each time a concrete set of parameters is presented to its inputs. This methodology uses both 'computing with words' and rough-fuzzy principles for the automatic classification of the musical instrument. Such an approach was previously applied to automatic assessment of acoustical quality, bringing reliable results and is currently implemented to automatic musical timbre recognition.

5.3 Evaluation of Hearing Impairment Using Fuzzy Approach

5.3.1 Problem Overview

Communication is essential for a properly functioning society. Hearing disorders are often a cause of communication problems. They can affect quality of life of persons with hearing loss. That is why proper fitting of a hearing aid is a very important part of the recovery process for people with hearing problems. However, adequate fitting of a hearing aid depends on the experience of the patient's doctor as well as the capabilities of the testing equipment which enable audiologists to adjust the hearing aid. On the other hand, computer technology makes it practical to organize such tests based entirely on computer software.

The proposed hearing aid fitting strategy employs a computer-based system. The system was developed in collaboration between the Warsaw-based Institute of Physiology and Pathology and the Gdansk University of Technology (Czyzewski et al, 2002a, 2002b, Kostek et al 2001, Skarzynski et al 2002). The project involves a systemic approach, and

needed the cooperation of people from information technology, sound engineers, medical doctors, and audiologists. During several years many projects developed and realized resulted from the cooperation. Knowledge, understanding and practical experience were gained by the author through this collaborative work.

A potential user of the system starts with the examination of loudness-growth characteristics in 4 frequency bands (a so-called Loudness Growth in ½ Octave Bands - LGOB procedure performed for 500, 1000, 2000 and 4000 Hz) (Allen et al 1990). In the loudness growth test a patient listens to the sound (a narrow-band noise) and the task is to evaluate his/her impression of loudness using such expressions as: too loud, very loud, loud, comfortable, soft, very soft, too soft. Sounds of various levels are presented to the patient in a random order, and each sound sample is presented to the patient a few times in order to eliminate mistakes. Then, these subjective responses are translated to levels in decibels (dB). An example of characteristics of a person with a hearing loss can be seen in Fig. 5.20). These characteristics are especially important in the case of patients with sensorineural hearing loss, because increasing hearing aid gain up to the amount of loss results in too much amplification. This is because of the much lower dynamics in the impaired hearing sense. In short, lower dynamics means that a person with a hearing loss after linear amplification doesn't hear soft sounds, whereas sounds of a moderate level are perceived as very low ones, and loud sounds may be perceived as too loud. On the other hand, typical responses of a person with normal hearing will be such as in Fig. 5.20a, but lying approximately on the diagonal of the diagram.

Based on these characteristics it is possible to generally classify the case of hearing impairment represented by a given patient. These characteristics allow also finding the shape of proper compressor characteristics. This situation is illustrated in Fig. 5.20b. Characteristics obtained in this way are used by the proposed system to simulate needed hearing aid performance. However, the standard method of measuring loudness growth characteristics employs filtered noise, whereas only the understanding of speech amidst noise can provide a final criterion for proper hearing aid fitting. Unfortunately, there are no means of direct mapping of standard loudness growth characteristics measured by noise to the characteristics corresponding to compressed noisy speech understanding. That is why the empirical testing procedure should also employ assessment of the level of understanding of speech patterns processed using adequately diversified compression curves (see the dashed lines in Fig. 5.20b) The diversification of these curves can be decided by the system. The interest region for diversifying these curves is defined according to the evaluated degree of the hearing impairment. The principle of this evaluation can be as simple as that:

the deeper the impairment, the wider is the interest region of diversified compression curves. However in the system engineered the interest region is established employing a fuzzy logic-based reasoning (Kosko 1992, 1997; Zadeh 1965, 1999b) and a so-called 'computing with words' approach (Czyzewski and Kostek 2002; Czyzewski et al 2000, 2002a, 2002b; Kostek and Czyzewski 2001a, 2001b, 2001c; Kostek et al 2004).

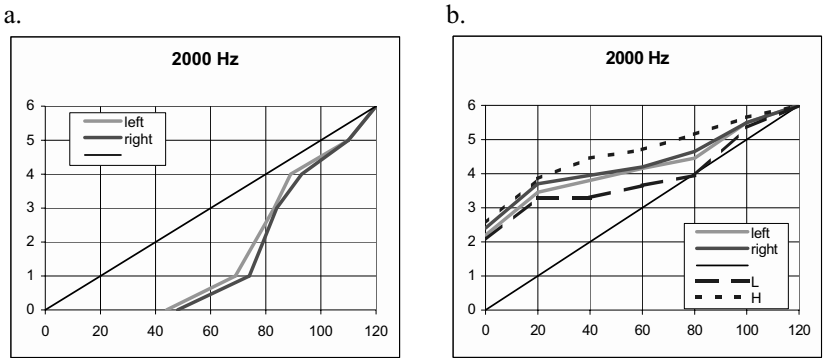


Fig. 5.20. Examples of loudness impression testing results for left and right ear representing the expansion characteristics (a) and reciprocal characteristics constituting the compression curves that should be used for the processing of sound in adequate hearing aids (b). The interest region of compression curves is also marked in dashed lines. *X*-axes represent sound level, *Y*-axes reflect the subjective loudness level

Characteristics, such as seen in Fig. 5.20b are used by the system MHAFS to simulate the needed hearing aid performance. The algorithm of the simulation is shown in Fig. 5.21.

It should be mentioned that apart from the computer a pair of properly calibrated earphones is used in testing. The earphones of choice are insert-type models. This is to partially overcome the problem of impedance mismatches between the artificial cavity of the headphones and the ear. These earphones and the computer sound interface were calibrated using the artificial-ear setup.

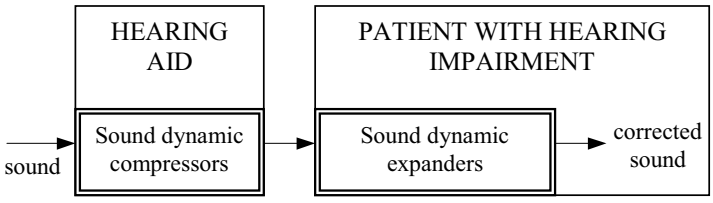


Fig. 5.21. Audio signal correction with hearing aid

In this Section, first, limitations of the clinical hearing aid fitting process are described. The audiological assessment in this process is based both on classical methods that use as a basis results of the audiometric test and the loudness scaling method. These methods employ artificial test signals. However, fitting of hearing aids should be also performed on the basis of testing speech understanding in noise, because this is much closer to the real life conditions. A satisfying reliability of these tests may be achieved through the use of modern computer technology with application of a properly calibrated sound system.

Several existing limitations in the clinical process of fitting hearing aids are, paradoxically, mainly due to the fast technology changes in the hearing aid field. One can observe not only a change from analog to digital technology but also the miniaturization of hearing devices, improved speech signal processing, better directional characteristics, etc. On the other hand the fitting process and the follow up procedures typically remain the same as previously used, thus more sophisticated methods are needed. In addition, clinical assessment uses artificial signals, so this process is far from the expected every day exploitation of a hearing aid. Other limitations were pointed out by Nowosielski in his paper (1997). The audiological assessment is very often performed using headphones. In this case one should take into account the impedance mismatch between the artificial cavity of the headphones and the patient's ear, because the accuracy of the hearing aid fitting is then limited. The lack of accuracy may also happen in cases when the direct monitoring of the hearing output in the patient ear canal is difficult. For example the insertion of the monitoring tube along the earmold is not possible due to the narrow ear canal or while inserted its presence affects parameters of the acoustic coupling between the hearing aid and the patient's ear or causes the acoustic feedback. As a solution to the mentioned problems Nowosielski proposed placing a subminiature microphone in the ear canal for measuring the air pressure close to the tympanic membrane during the fitting process. This improvement was left however to further development (Nowosielski 1997). The fitting procedures are also long and tiring for a patient. Therefore there is a continuous need to develop new strategies in hearing aid fitting procedures and the supporting technology. A satisfying fitting strategy reliability can be achieved through the use of modern computer technology with application of a properly calibrated sound system. To partially overcome the problem of impedance mismatch between the artificial cavity of headphones and the ear, the inserted earphones can be used in such a system.

The objective of this Section is also to show fuzzy logic-based dynamic assessment of hearing prostheses (Kostek and Czyzewski 2001a, 2001b, 2001c; Czyzewski and Kostek 2002). Some issues shown in this Section

were developed by P. Suchomski (2005), who under the guidance of the author, is now submitting his Ph.D. to the Scientific Council of the Faculty of Electronics, Telecommunications and Informatics, GUT.

5.3.2 Multimedia Hearing Aid Fitting System (MFHAS)

The developed software is provided with a multimedia interface in which elements of graphics and computer animation are used. The system engine is based on fuzzy logic principles. The system role is to estimate characteristics of hearing sensitivity of the patient and to provide an approximate diagnosis of the degree of hearing impairment for this patient. The lay-out of the algorithm of the engineered system is presented in Fig. 5.22 and its multimedia interface design is showed in Fig. 5.23.

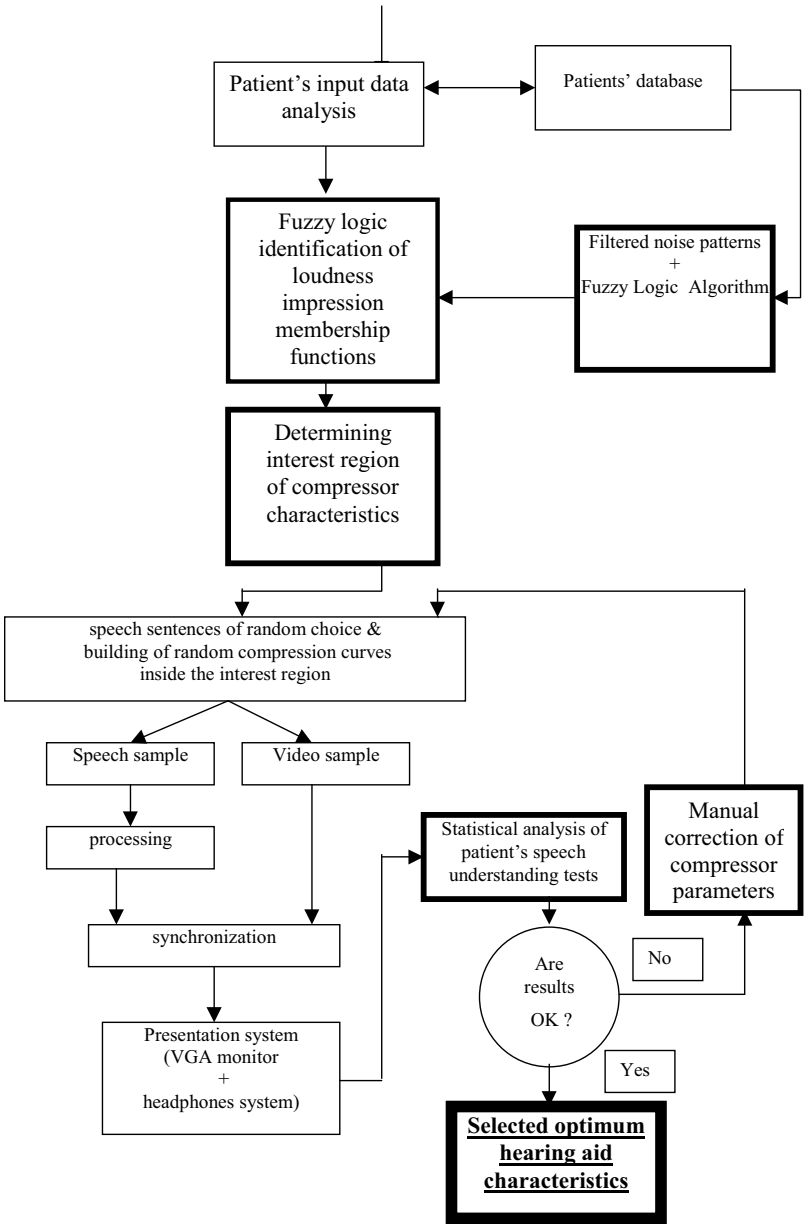


Fig. 5.22. MHAFS algorithmic diagram

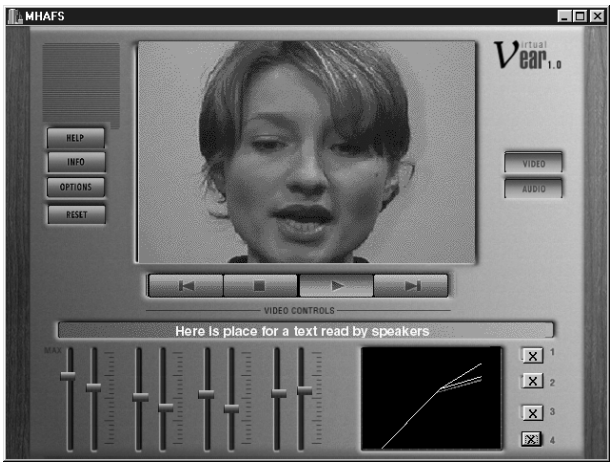


Fig. 5.23. User interface design. In the lower part of the computer screen sliders are seen which make it possible to change the shape of compression curves within the interest region

5.3.3 Evaluation of Hearing Impairment Using Fuzzy Approach

Identifying Loudness Membership Functions

In general, in order to determine the dynamics characteristics of a hearing aid on the basis of the results of loudness scaling one calculates the difference in dB between the result for impaired hearing and the corresponding result for normal hearing. Such procedure results from the Stevens' theorem, which assumes that the loudness perception of an impaired person is equal to that of a normal-hearing person for different sound levels. For example, if a hearing-impaired person evaluates the loudness perception of a 60dB SPL test signal in the category 'VERY SOFT', he/she feels the same loudness perception as a normal-hearing person does in the case of a 40dB SPL test signal. The resulting difference of 20dB means, that in order to compensate for the hearing loss the hearing aid should amplify the signal by 20dB. On the other hand, in order to model the hearing dynamics of a hearing-impaired person one should lower the test signal by 20dB.

The first problem is to construct the loudness-scaling characteristics for normal hearing. For the results to be statistically reliable, one would have to test several dozens of normal-hearing persons. Another problem is to properly calculate the difference between the results of the given loudness-scaling test and the averaged results for normal hearing.

Most audiologists determine the preliminary characteristics of the hearing aid using a simple calculation procedure. Experience shows that such a method does not guarantee finding optimum settings of the compression circuits, but it allows assessing the characteristics of the searched hearing aid in a relatively straightforward and intuitive way. This technique requires additional tuning of the determined characteristics using other methods of adjusting hearing aids. The difficulty in determining hearing characteristics on the basis of LGOB test results lies primarily in converting the subjective scale of loudness perception into the objective scale of sound level expressed in dB. The widely used method of determining hearing characteristics described above implicitly projects the subjective scale of categories onto the space of real numbers from the closed range from 0 to 6, and subsequently calculates the difference between the results of loudness scaling for normal hearing and those for the tested case. This problem may alternatively be solved using a method that would in a natural way convert the results of the loudness-scaling test in the category domain, i.e. would determine the difference between normal and impaired loudness scaling in a way similar to that of a human expert, using a set of categories like e.g. very small difference, small difference, medium difference, big difference and very big difference. Subsequently, a proper interpretation of these categories would be required to determine the correct sound level in the dB SPL scale. These requirements are met by a method employing fuzzy logic-based processing.

For a fuzzy-logic system to determine the static characteristics of hearing dynamics on the basis of LGOB test results, the following information is required:

- Frequency and sound level of subsequent test signals;
- Data describing correct loudness scaling with the LGOB test;
- Results of the given LGOB test;
- Knowledge on interpreting the differences between the analyzed results and those for normal hearing;
- A method of calculating the difference in dB for the analyzed LGOB test result.

Fuzzification of Input Parameters

In the case of a typical fuzzy logic system the first stage of processing is a so-called fuzzification. In the presented method two input parameters need to be fuzzified: the level of the test signal being processed (expressed in dB) and the LGOB test result. It should be remembered that the process of fuzzifying the input signal level depends on the signal frequency. In the

described case seven membership functions are required for every frequency band. Every function describes the distribution of the given category of loudness perception depending on the test signal level. As there are four frequency bands of center frequencies of 500Hz, 1000Hz, 2000Hz and 4000Hz tested in the LGOB test (Fig. 5.24), four sets of membership functions are required. For preparing such membership function one has to perform the LGOB test on several dozens of normal-hearing persons. This was done during laboratory sessions at the Multimedia Systems Department. Over 80 students with normal hearing were tested during the academic year, this means that over 160 test results were obtained for the analysis. During the test the system generates some hundreds of samples for each frequency band. A subject, while listening, is instructed to rate his/hers judgements using such labels as ‘too_soft' (or numerically 0), ‘very_soft’ (1), ‘soft’ (2), ‘comfortable’ (MCL – most comfortable level) (3), ‘loud’ (4), ‘very_loud’ (5) and ‘too_loud’ (UCL – uncomfortable level due to high loudness) (6).

The membership functions obtained on the basis of a generally accepted approximation of LGOB test results for normal hearing is shown in Fig. 5.25. Analysis of the expected plots of LGOB test for normal hearing reveals that they are identical for each analyzed frequency band. This is an assumed simplification resulting from both audiology experience and the analysis of Equal Loudness Curves, where one may notice that differences between equal hearing curves in the analyzed bands do not exceed 10dB.

A membership function describes the distribution of the membership degrees of elements of the given type to the given fuzzy set. Typically they are trapezoidal- or, in particular cases, triangle-shaped; in less frequent cases they are shaped after sigmoidal functions or bell-shaped curves. The tendency of using trapezoids or triangles for approximation results primarily from practical considerations.

Membership functions are designed on the basis of expertise. It can come from an expert in the given field or result from statistical analysis of the given phenomenon. In the case of of statistical analysis one has to determine the statistical distribution of the given phenomenon or assume the normal distribution (if probability distribution is unknown), and then to assess the minimum set of tests necessary to create a membership function which would be maximally consistent with the factual probability distribution of the analyses variable (test of distribution compatibility).

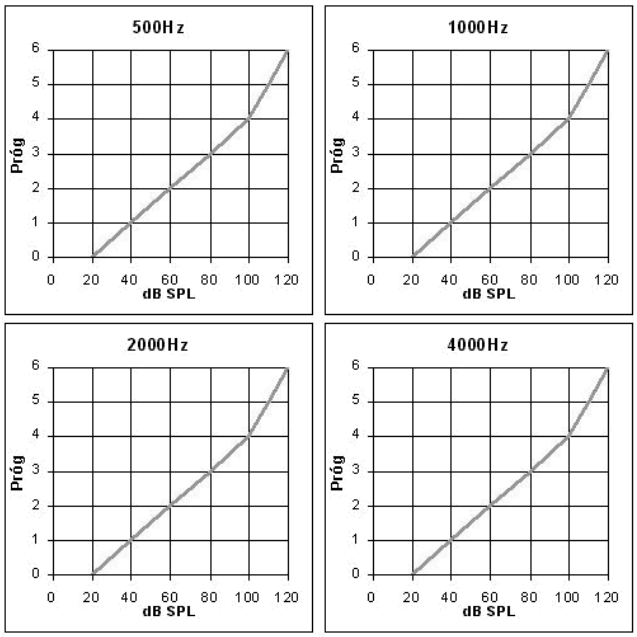


Fig. 5.24. Results of the LGOB test for a normal hearing (*X*-axis represents sound level, *Y*-axis reflects subjective loudness level

Consecutive Membership Function (MF) shown in Fig. 5.25 correspond to the subjective categories presented before. There are seven categories of loudness perception and each of them constitutes one fuzzy set, therefore one has to generate seven membership functions, one for each set. The method of generating membership function plots can be best explained based on an example.

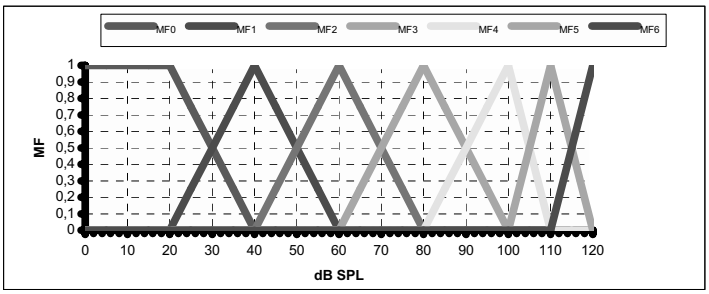


Fig. 5.25. Membership functions based on loudness scaling test (from the left side labels reflects MF0, MF1, MF2, MF3, MF4, MF5, MF6)

For example, a membership function describing a fuzzy set representing the ‘MCL’ category of loudness perception evaluation would take the maximum value (the maximum degree of membership), that is 1, for the sound level of 80dB SPL (according to the conventional curve describing the LGOB test for normal hearing this exact level should be interpreted as ‘MCL’, Fig. 5.25). This function would take the value of 0 for sound levels equal to or greater than 100dB SPL as well as for those equal to or lower than 60dB SPL. In this fashion one would define a triangle-shaped function. In an analogous fashion one can create functions for the other categories of loudness perception.

From the statistical point of view one should assess the minimum number of required tests. As in most cases the determined membership functions and the expected ones have the same shape and range, one can assess the number of required tests by analyzing one of the functions, e.g. MF3 (the one describing the fuzzy set associated with the ‘MCL’ category of loudness perception). This function covers the range from 60dB SPL to 100dB SPL, in which there are seven values of test signal level (investigated with the step of 5dB) with the membership degree to the ‘MCL’ set different from 0. Therefore this range can be split into seven equal subranges by dividing the area under the function curve into seven parts (Fig. 5.26).

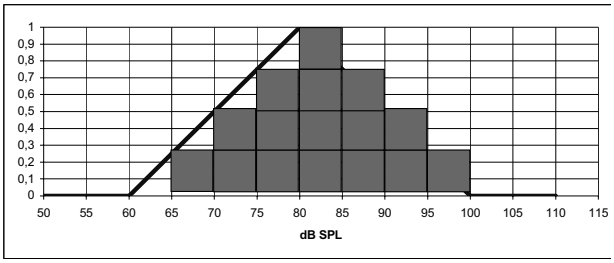


Fig. 5.26. Assessing a number of required tests

The minimum number of plots required for reliable determination of the desired probability distribution can be determined on the basis of the chi-square test results, which describe consistence concerning probability distribution of the given random variable with its factual distribution.

$$\chi^2_{k-1} = \sum_{i=1}^{k} \frac{(n_i - N \cdot p_i)^2}{N \cdot p_i} \tag{5.32}$$

where the denotations are as follows:
ξ – random variable
F – cumulative distribution function of random variable
$F(a) = 0$, $F(b) = 1$, $a = a_0 < a_1 < a_2 < \ldots < a_k = b$ and $p_i = P\{a_i\text{-}1 < \xi <= a_i\}$, i=1,2,...
n_i - number of elements ξ fulfilling the condition $a_i\text{-}1 < \xi <= a_i$ (observed size in range i);

$p_i = \int_{r_i}^{r_{ii+1}} p(x)dx$ - probability that the random variable X fits within range i

r_i – lower boundary of range i
N – number of all observations.

The chi-square test can be used to verify the hypothesis concerning random distribution of random variable ξ only when N is sufficiently large. It is assumed that the use of this test is justified when for every j e.g. j>=10 or n_j>=5. In practice this condition means that the number of tests to be performed for a random variable taking lowest values in the given range should at least be greater than or equal to 5. In the case illustrated in Fig. 5.26 the random variable takes the lowest values at 65dB and 95dB. The number of tests used for determining the distribution value in this point can be expressed as the area of rectangle covering the neighborhood of the analyzed point in the given distribution. This means that the area of this rectangle should be greater than or equal to 5. In the analyzed distribution one can define as many as 16 such rectangles. When looking for the minimum number of tests required for determining a distribution of such type, one should assume that the area of a single rectangle equals 5 and therefore the sum of areas of all rectangles equals 80. This number can be assumed as the minimum number of observations required for reliable determination of the distribution of the given random variable. The above considerations result in a conclusion that in order to obtain a set of reliable membership functions one has to perform the LGOB test on at least 80 normal-hearing persons.

One of the basic methods of approximation of membership functions is the approximation with triangle-shaped functions. It can be performed on the basis of an ordinary mean-square approximation. For this aim one should determine equations of two straight lines approximating the triangle sides in the mean-square sense. The algorithm used to determine the triangular membership functions involves the following steps:

- Finding the value of the first element of the given fuzzy set (the value of the first argument, for which the factual membership function takes a non-zero value)
- Considering all the elements of the factual membership function *MF* fulfilling the equation given below, in order to determine the first side of the triangle:

$$x : \left\langle \underset{x_i}{\forall} \left(MF(x_i) - MF(x_{i-1}) \right) > 0 \right\rangle \tag{5.33}$$

  where i – subsequent indices of arguments of membership functions *MF* fulfilling condition (5.33),
- Calculating parameters a_1 and b_1 of the line $y=a_1x+b_1$,
- Considering all the elements of the factual membership function *MF* fulfilling the equation given below, in order to determine the second side of the triangle:

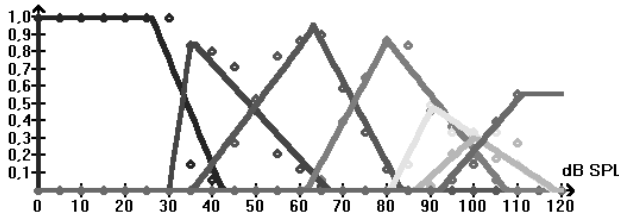
$$x : \left\langle \underset{x_i}{\forall} \left(MF(x_i) - MF(x_{i-1}) \right) <= 0 \right\rangle \tag{5.34}$$

  where i = subsequent indices of arguments of membership functions *MF* fulfilling condition (3),
- Calculating parameters a_2 and b_2 of straight line $y=a_2x+b_2$,
- Calculating the point of intersection of straight lines $y=a_1x+b_1$ and $y=a_2x+b_2$ (determining the triangle vertex),
- Calculating zeros of both lines.

An example of a set of membership functions for the frequency band of 500Hz obtained by approximating the factual values of membership functions with triangles is illustrated in Fig. 5.27a. In this case individual elements may belong to more than two fuzzy sets, thus further fuzzy logic-based processing is more complicated. Membership functions, which share a part of their domain with domains of other membership functions (intersection with more than two other fuzzy sets), do not have the maximum value equal to 1 and corresponding to the maximum degree of membership of the given element to the given fuzzy set. This means that in reality there are no arguments of the membership function which would belong only to this function. It turns out that such situation is only possible if membership functions are determined on the basis of the averaged results of loudness scaling. Only then each fuzzy set "neighbors" (intersects) at most two other fuzzy sets and there are elements, for which the average value of loudness scaling results points directly to a given category of loudness perception evaluation. In reality the situation when the whole population of

normal-hearing persons would evaluate the hearing perception of a given test signal level as exactly the same does not happen. Since in fuzzy processing using functions that reach the maximum membership value of 1 is recommended, in the discussed case one needs to normalize each membership function to the maximum value (Fig. 5.27b).

a.



b.

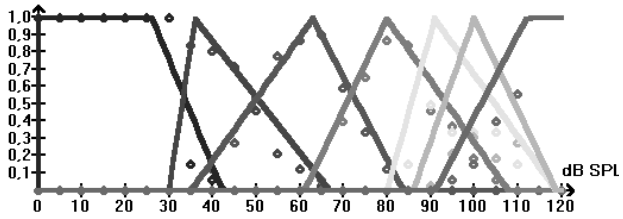


Fig. 5.27. Approximation of the factual values of membership functions with triangles, a) without normalization, b) with normalization

Construction of Membership Functions Corresponding to Loudness Perception

In general, if the exact statistical distribution of the investigated phenomenon is not known, one assumes that it is governed by the normal distribution. Although in practical fuzzy-logic applications membership functions described with Gaussian curves are sometimes used, they are not convenient to analyze. The problem lies in the fact that a Gaussian function does not reach the value of 0, which in practice means that all analyzed sets intersect, and therefore each element belongs to all fuzzy sets. This problem can be solved in two ways. One involves defining a threshold value, below which the Gaussian function value would be treated as zero. The other is based on replacing the Gaussian function with one that approximates it better. The easiest function approaching the Gaussian function is the quadratic function (parabola) that is a trinomial square in the form $y=ax^2+bx+c$.

It can be determined on the basis of mean-square approximation with a second-degree polynomial. The formula for trinomial square coefficient can be worked out by solving the orthonormal set of equations, similar as in the case of straight lines.

Fig. 5.28 presents a set of membership functions derived by approximating the factual values of membership functions describing loudness scaling for test signals in the 500Hz band. However, comparing the mean square errors for approximation with triangles and parabolas reveals that the approximation error is smaller in the case of triangles than in that of quadratic functions.

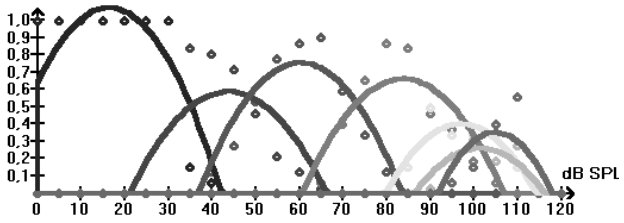


Fig. 5.28. Approximation of the factual values of membership functions with quadratic functions

Assuming that a membership function has the properties of a probability density function and therefore the expression (5.35) is fulfilled, one can approximate the factual values of membership functions with trapezoid distributions.

$$\int_{-\infty}^{+\infty} p(x) = \int_{a}^{d} p(x) = 1 \tag{5.35}$$

where $p(x)$ is probability density function.

Fig. 5.29 presents an example result of approximating the factual values of membership functions in 500 Hz frequency band with trapezoid functions.

One can see that the test indicates greater consistence of trapezoid functions, which means that trapezoid distributions are a better choice than other shapes of *MF*.

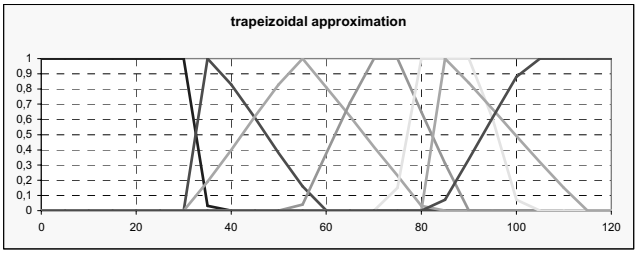


Fig. 5.29. Example of approximation of membership functions with trapezoid functions (from the left side labels have the meaning – 'too soft', 'very soft', 'soft', 'MCL', 'loud', 'very loud', 'UCL')

Besides the information on loudness scaling for normal-hearing persons, the fuzzy logic system being designed requires information of loudness scaling results for the investigated person. In the case of the first parameter fuzzification involves determining degrees of membership to individual fuzzy sets (categories of loudness perception evaluation) for the given test signal level expressed in dB. Such a procedure is only possible on the basis of the designed membership functions. In the case of the second parameter such fuzzification procedure is impossible, as membership functions describing loudness scaling for the investigated person are not known. The exact evaluation of loudness perception chosen by this person during the LGOB test for the given test signal level is known, however. In this case fuzzification can be performed by a so-called singleton method. In this case fuzzification takes into account the fact that the loudness perception evaluations given by the investigated person takes the value of 1, that is the maximum value of the degree of membership to the fuzzy set corresponding to the given category. At the same time the membership degrees related to other fuzzy sets take the values of zero.

To sum up, the presented system requires two parameters at input: one is the test signal level for which loudness perception is being evaluated, the other is the category used by the investigated person for this test signal level. Next, the process of fuzzification allows determining degrees of membership of the given test signal level to each of seven fuzzy sets, described with membership functions designed on the basis of analysis of LGOB test results for normal hearing. In the case of the second parameter the corresponding fuzzy sets (seven as well, since there are seven categories of loudness perception evaluation) are one-element sets, therefore in this case a given element can belong to only one of the seven fuzzy sets and the degree of membership to the given set equals 1.

Input Functions of the Designed Fuzzy System

In the previous paragraph the process of determining membership functions and fuzzification of input parameters was described. The next step involves defining the system output. As described earlier, the aim of the designed method is determining the dynamics of impaired hearing. This means that the designed system should "calculate" the difference between the given loudness perception evaluation and the correct loudness perception evaluation corresponding to the given test signal. This difference should be expressed in dB.

Analysis of a typical plot of LGOB test results reveals that between seven categories of loudness perception evaluation one can define six differences pointing to hearing loss (area below the LGOB curve for normal hearing) and six differences pointing to hypersensitivity (area above the LGOB curve for normal hearing). Zero difference is a special case of difference. The above analysis leads to a conclusion that the output of the described fuzzy system can be described by a set of thirteen membership functions (Fig. 5.30) expressing the difference between the evaluation of factual loudness perception and the evaluation for normal hearing. Fuzzy sets obtained in this fashion can be described with the following labels (describing the difference size): the MF in the middle of Fig. 5.30 is related to the label: 'none', then to the right there are the following lables: 'very small', 'very small+', 'small', 'small+', 'medium', 'medium+', 'large', 'large+', 'very large', 'very large+', 'total', 'total+'. Labels marked with '+' sign denote positive difference (hypersensitivity). From the mid MF to the left the assigned labels refer to negative difference.

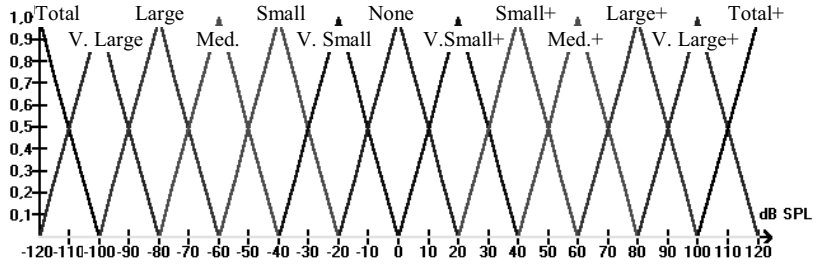


Fig. 5.30. System output: from the middle of the figure to the right the following labels are assigned: none, very small, very small+, small, small+, medium, medium+, large, large +, very large, very large +, total, total+, to the left differences are negative

Rule Basis

Fuzzy processing depends on properly defined rule basis. Fuzzy logic rules have the following form:

If <premise1> AND <premise2>AND...<premise_n> THEN decision

In the discussed case there are two premises. One is associated with the information on normal loudness scaling; in further considerations it is denoted as the **Norm** variable. The other premise is associated with the investigated results of LGOB test; it is denoted as the **Exam** variable. Since both premises apply to the results of the LGOB test, they both use the same categories describing loudness perception. In order to differentiate the fuzzy sets associated with individual premises, labels of fuzzy sets associated with the first premise use lower case letters while those of fuzzy sets associated with the second premise utilize upper case letters.

In general, the rule basis is designed on the basis of expertise. In this case such expertise can be derived from analysis of the LGOB test. Analysis of LGOB test results for normal-hearing persons showed that the phenomenon of loudness scaling is linear in character, however the factor of proportionality rises from 1:1 to 2:1 (the loudness perception rises twice faster) for test signals of levels exceeding 100dB SPL. On the basis of this information one can design a rule basis according to the following guidelines:

- Premises pointing to consistence of loudness perception evaluation for normal loudness scaling and for the investigated loudness scaling generate a decision stating no scaling differences and marked with the label none.

  e.g.: IF Norm is well AND Exam is WELL THEN d is none

- If the given result of loudness scaling differs by one category of loudness perception evaluation, the decision is associated with the output labeled ‘very small’ in the case of negative difference or very ‘small+’ for positive difference.

  e.g.: IF Norm is well AND Exam is SOFT THEN d is very small
  IF Norm is well AND Exam is LOUD THEN d is very small+

- If the given result of loudness scaling differs by two categories of loudness perception evaluation, the decision is associated with the output labeled ‘small’ in the case of negative difference or ‘small+’ for positive difference.

  .....

- If the given result of loudness scaling differs by six categories of loudness perception evaluation, the decision is associated with the output labeled ‘total’ in the case of negative difference or ‘total+’ for positive difference.

On the basis of the above principles one can build a complete rule basis.

Defuzzification

The last stage of designing any fuzzy logic system is choosing a method of defuzzification, i.e. a method of converting the obtained categories to numerical values. The main difficulty at this stage is that in the analyzed phenomenon the factor of proportionality significantly increases above 100dB SPL, the change must be accounted for in the defuzzification process. The modification involves dividing the obtained difference for levels above 100dB SPL by the proportionality factor for this range of sound levels, in this case 2. This property results in another consequence that has to be taken into consideration during defuzzification, namely the ‘distance’ between the ‘very loud’ and ‘too loud’ categories is also twice smaller than between the other categories. That is why the difference obtained during evaluation of loudness perception for levels lower than or equal to 100dB SPL for the very loud category should be decreased by 10dB, while for the too loud category it should be decreased by 20dB.

Fuzzy Logic-based Algorithm for Determining Hearing Dynamics

The previous paragraphs describe the design details of a fuzzy logic system, which determines the difference between the currently analyzed loudness scaling and the normal loudness scaling. In order to determine the whole dynamic characteristics of the investigated hearing one should create an algorithm, which would calculate the desired hearing dynamics characteristics on the basis of LGOB test results using the designed method of determining the difference between normal and impaired loudness scaling. Fig. 5.31 presents the scheme of the whole module, which accepts the stream of results of the given LGOB test at its input and produces a stream of subsequent differences for subsequent results of the LGOB test at its output.

At this point the LGOB test result is understood as three parameters (level, frequency, evaluation), where level is the level of the given test signal (expressed in dB), frequency is the frequency band encompassing the given test signal (expressed in Hz), while evaluation is the loudness perception category used to evaluate the loudness perception caused by the

given test signal (expressed as an index of loudness perception evaluation category – from the range of 0–6).

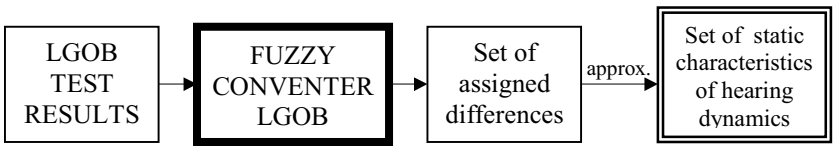


Fig. 5.31. Diagram of fuzzy processing-based algorithm for determining static characteristics of hearing dynamics

At this stage of description of the designed method one should notice that the determined differences (both in the classic and in the designed method) are not really identical with subsequent points of the hearing dynamics characteristics being determined. Determined differences can be interpreted geometrically as the distance of the obtained result from the curve describing the averaged results of LGOB test for normal hearing. That means that in order to determine the desired static characteristics of hearing dynamics one shall approximate the obtained results.

Fig. 5.32 presents an example plot of loudness scaling results with the LGOB test, which were obtained for test signals from the frequency band centered around the frequency of 500Hz. The analysis of results of the LGOB test (Fig. 5.32) reveals that the quietest test signal evaluated in the analyzed band had the level of 35dB SPL (read from the plot), while the loudest signal reached the level of 110dB SPL.

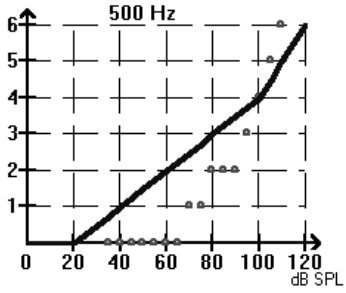


Fig. 5.32. Sample LGOB results (points in the characteristics)

In the LGOB test, the test signal changes with a step of 5dB. Bearing in mind the above information, one can perform the following analysis:

- The signal with the level of 35dB SPL was evaluated within category 0, i.e. TOO SOFT. According to the standard method the difference for this result compared with the normal results equals to 15dB (as cate-

gory 0 is assigned to the level of 20dB SPL), therefore one should assume the output level of 20dB for the input level of 35dB on the static characteristics of hearing dynamics (35dB – 15dB = 20dB).

In the fuzzy LGOB converter a 35dB SPL signal belongs to two fuzzy sets; membership degree to the too soft set equals 0.25, while membership degree to the very quiet set equals 0.75 Two rules are activated:

if Norm is too soft AND Exam is TOO SOFT THEN d is none
min(0.25,1) = 0.25
IF Norm is very quiet AND Exam is TOO SOFT THEN d is very small
min(0.75,1) = 0.75

Defuzzification determines the difference of -14.21dB and from this one can deduct that output level of 20.79dB should be assumed on the static characteristics of hearing dynamics for the input level of 35dB SPL (35dB – 14,21dB = 20,79dB).

…….

- The signal with the level of 110dB SPL was evaluated within category 6, i.e. TOO LOUD. According to the standard method the difference for this result compared with the normal results equals +10dB (as category 6 is assigned to the level of 120dB SPL), therefore one should assume the output level of 120dB for the input level of 110dB on the static characteristics of hearing dynamics (110dB + 10dB = 120dB).

In the fuzzy LGOB converter a 35dB SPL signal belongs to two fuzzy sets (fuzzification); membership degree to the very loud set equals 1. The following rule is activated:

if Norm is very loud AND Exam is TOO LOUD THEN d is very small+
min(1,1) = 1

Defuzzification determines the difference of +10dB and from this one can deduct that output level of 120dB should be assumed on the static characteristics of hearing dynamics for the input level of 110dB SPL (110dB + 10dB = 120dB).

The presented example shows that differences between results obtained according to both methods are negligible. As sound levels below 20dB SPL are treated as inaudible in the loudness scaling test (in the designed computer version of the loudness scaling test this threshold is set to 30dB), for increased plot clarity results of value lower than or equal to 20dB are treated as equal to 0dB for the needs of approximation.

The designed method allows for processing descriptive data (scale of loudness perception). Moreover, this method uses all the statistical knowl-

edge on proper loudness scaling with the LGOB test, not only the average values as is the case with the standard method.

From the above given discussion it can be seen that the approximation of the membership functions corresponding to hearing perception is not a trivial case.

The designed computer LGOB test enables to obtain results consistent with the results of loudness scaling registered with a professional device dedicated for clinical tests. Designing the computer LGOB test forms the basis for creating an open and elastic computer utility for diagnosing hearing, determining optimum characteristics of desired hearing aids as well as for performing coarse simulations of both hearing loss and the desired hearing aid.

Speech Pattern Testing

After the loudness impression characteristics are obtained for a given patient, and processed by the fuzzy logic engine, the system performs speech pattern testing. Speech signal is passing through four partially overlapping filter bands with the following middle frequencies: 500, 1000, 2000 and 4000 Hz. The signal dynamics are modeled in each band on the basis of sound compression characteristics. The processed signal is played back into the patient's headphones. The system stores 600 phonetically balanced audio-video recordings of simple sentences based on colloquial language. They are read partly by a female and a male speakers. The patient listens to the recordings randomly chosen by the system during the test. The system shows synchronized video recordings of speakers' faces. This feature is needed for deeply hearing impaired patients who are capable of lipreading.

After a single recording is played back and then received by the patient, the text of the sentence is shown on the screen. The patient self-estimates the level of understanding of the recording just played back using the established subjective assessment scale. Once the tests are completed the system analyzes the scores assigned by the patient to individual patterns played back.

On the basis of the results the system presents optimized dynamic characteristics of the hearing aid matching the patient's needs.

Concluding Remarks

The system for testing hearing with the use of an expert multimedia system may be helpful to properly diagnose patients and to give them some kind of sound experience before the hearing aid is selected for them. The hearing characteristics are assessed using the modified loudness scaling test.

Since the compression curves derived from testing with filtered noise usually differ from compressor settings desired for optimal speech understanding, a special procedure has been established that enables to find a region of interest for testing compression characteristics with processed speech patterns. Consequently, this region of interest is determined on the basis of an extended loudness scaling test. The modification of the hearing aid fitting procedure lies in the introduction of fuzzy logic principles to the processing of results of testing loudness impression with filtered noise samples. The fuzzy processing of patients' responses employs membership functions identified by normally hearing population and in this way the degree of impairment for an individual is discovered. The proper compression characteristics that should be used in the hearing aid of a concrete patient are tested finally by speech patterns in order to optimize further speech understanding.

5.4 Neural Network-Based Beamforming

5.4.1 Problem Overview

Theoretical bases of beamforming and spatial filtration have already been shown in Chapter 4.1. Automatic identification of sound sources direction is however still an unsolved problem in many real-life applications, such as for example, hearing prostheses or contemporary teleconferencing systems. In many situations people have difficulty in understanding speech in surroundings with background noise, high reverberation and/or with many concurrent speakers. This is often called the “cocktail-party-effect”. Speech signals coming from various directions not only interfere with the target signal but also can obscure it. One approach to reducing this noise is to provide directional field of hearing. Sounds coming from sides and back are attenuated while sounds coming from front are left without attenuation. Source identification system should allow tracking a target speaker automatically without much delay in order to avoid picking up concurrent speakers by the same microphone channel. This may be done in various ways, however, generally two approaches can be found in literature. As mentioned in Chapter 4.1, one of them is a classical approach to this problem based on delay-summation algorithms, superdirective arrays and adaptive algorithms, non-linear frequency domain microphone array beamformers, etc. (Adkins and Turtora 1996; Berdugo et al 1999; Brandstein 1997; Chern and Lin 1994; Frost 1972; Griffiths and Jim 1982; Widrow

and Stearns 1985). The effectiveness of these algorithms was diminishing while performing in reverberant environments. Examples of such algorithms were reviewed in Chapter 4.1. The second solution to this problem was proposed in Multimedia Systems Department, GUT in collaboration with the Institute of Physiology and Pathology of Hearing, Warsaw in previous studies, namely Artificial Neural Networks (ANNs) for the purpose of the automatic sound source localization have been applied (Czyzewski 2003a, 2003b; Czyzewski and Lasecki 1999; Kostek et al 1999; Lasecki and Czyzewski 1999; Lasecki et al 1998, 1999).

In the study shown below two types of ANNs were used, namely feedforward neural networks and recurrent ones. They both differ in ANNs structures and properties and feature vectors that are fed to the given algorithm input. The ANNs were used as nonlinear filter in the frequency domains or only in the time domain.

A series of experiments was organized employing ANNs for the automatic detection of sound source position in a noisy acoustical environment. In addition a comparison of results obtained using both standard feedforward ANNs and RNNs are presented. Some details concerning the implementation of the proposed algorithms are described. On the basis of the experiments carried out some conclusions were drawn out and further developments of the neural network-based spatial filter were discussed in order to use it in teleconferencing.

These studies were conducted for several years, sponsored by the State Committee for Scientific Research, Warsaw, Poland. Some colleagues from the Multimedia Systems Department research team took part in them. A very valuable input was done by Czyzewski, Lasecki and Krolikowski (Czyzewski 2003a, 2003b; Czyzewski and Krolikowski 2001a, 2001b; Czyzewski et al 1998, 2001; Czyzewski and Lasecki 1999; Krolikowski et al 2001; Lasecki and Czyzewski 1999; Lasecki et al 1998, 1999). Also, Kosicki (2000), M.Sc. student, supervised by the author took part in these experiments.

Psychophysiology Background

Psychophysiology studies show that binaural perception is based on two principal acoustical cues (Bodden 1993; Hartmann 1999), namely *Interaural Level Difference* (ILD) and *Interaural Time Difference* (ITD). The first one refers to the difference of intensities of waveforms in the left and in right ears and the second one to the difference of arrival times, which is equivalent to the phase difference of the waveforms. There are some additional factors that influence sound localization perception. Phenomena underlying sound localization are not finally recognized, hence this is still a

subject of intense research. Sound direction is often described in terms of angles around the head (Fig. 5.33). In experiments only the horizontal plane was taken into account.

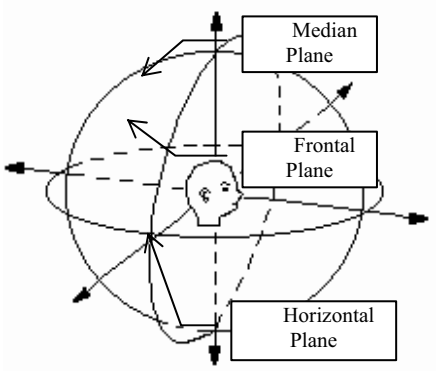


Fig. 5.33. Binaural sound perception

In real conditions various distortions such as background noise and reverberated sounds can occur that interfere with the target signal. By means of signal processing it is possible to model human localization perception using either linear or nonlinear approach (Khahil 1994; Mahieux et al 1996). Under the above formulated assumptions signals received by a linear microphone array are expressed by the following relationships:

$$\begin{cases} x_1(t) = \alpha_1 \cdot h_1(t) * s(t) + n_1(t) \\ x_2(t) = \alpha_2 \cdot h_2(t) * s(t-\tau) + n_2(t) \\ \quad \vdots \\ x_i(t) = \alpha_i \cdot h_i(t) * s(t-(i-1)\cdot\tau) + n_i(t) \\ \quad \vdots \end{cases} \tag{5.36}$$

where $h_i(t)$ is an impulse response of the reverberant channel associated with the ith microphone, and $n_i(t)$ denotes ambient noise received by the ith microphone.

Eq. (5.36) shows that the problem of sound source localization is very complex, and therefore a number of various methods have been proposed in literature (Berdugo et al 1999; Brandstein 1997; Chern and Lin 1994). These methods were also used in multimedia applications (Aoki and Okamoto 1999; Jacovitti and Scarano 1993; Kostek et al 1999; Mahieux et al 1996; Wang and Chu 1997).

5.4.2 Data Processing

In the present study experimental procedures were two-fold. First, sound material was recorded in a small studio. All sounds were monophonically registered with the sampling frequency equal to 44.1 Hz and 16bit/sample resolution. They were such as: lists containing 100 logatoms and some phrases. These were read both by female and male speakers. In addition white noise, filtered noise and tones were recorded. All processed sound files were then converted to 22.05 kHz stereo format and normalized to -6dB.

In addition, similar recordings were done in an anechoic chamber. A circular array consisted of 8 electret microphones set on the circumference of a 0.15 m radius rim and positioned 1.58 m above the floor was used. It was placed in the horizontal plane. Eight mono tracks were recorded simultaneously. The recording parameters were as follows: 16 bit/sample and the sampling frequency was equal to 48 kHz. There was one male speaker, distanced 1.5 m from the array. The speaker read a logatom list every 5^{o}. In result 72 eight-track recordings were made, and every recording lasted approximately 55 s.

The second step consisted in extracting feature vectors to be fed to the learning algorithms. During the parameterization process the signal was divided into frames of the length of 256, 512 or 1024 samples. Two sets of feature vectors were prepared. The first set was based on previously defined parameters under the assumption that a neural network provides an effective non-linear filtering algorithm of an acoustic signal transformed into the frequency-domain (Czyzewski et al 1998, Kostek et al 1999; Lasecki et al 1999). On the other hand since some of these parameters are not orthogonal, thus they may be eliminated from the feature vector and other orthogonal parameters can be formulated (Czyzewski et al 1999, 2001). This approach will be described later on.

It was assumed that the number of microphone channels has been limited to two. Signal arriving at both microphones can be written as:

$$l(t) = s(t) + n_l(t); r(t) = s(t) + n_r(t) \tag{5.37}$$

where:

- $l(t)$, $r(t)$ - signals received by the left and right microphones,
- s(t) - desired signal arriving from the front direction,
- $n_l(t)$, $n_r(t)$ – signals coming from the lateral or backward directions arriving to the left microphone and to the right microphone. These signals are treated as noise.

The main task of the spatial filter is to estimate the desired signal $s(t)$ arriving from the forward direction. It is neither desirable nor possible to completely attenuate signals from lateral and backward directions. Because spatial filter works in the frequency domain, it is assumed that each spectral component, which represents signals coming from unwanted directions should be attenuated by at least 40 dB (see Fig. 5.34). In Fig. 5.34 a prototype spatial characteristics is shown. The spectral components that represent signals coming from the forward direction should remain unchanged. This can be described by the following expressions:

$$\tilde{L}\left(e^{j\omega}\right)=\sum_{i=1}^{N} g(i)L^{i}\left(e^{j\omega}\right);\ \tilde{R}\left(e^{j\omega}\right)=\sum_{i=1}^{N} g(i)R^{i}\left(e^{j\omega}\right) \tag{5.38}$$

where:

- i – spectral component index,
- $\tilde{L}\left(e^{j\omega}\right),\ \tilde{R}\left(e^{j\omega}\right)$ - estimates of a signal component L^{i} in the left, and R^{i} in the right channel,

$g(i)$ – attenuation coefficient of noisy components described by the following formula:

$$g(i)=\begin{cases} 1; i \in \textit{Signal components} \\ 0.01; i \in \textit{Noise components} \end{cases} \tag{5.39}$$

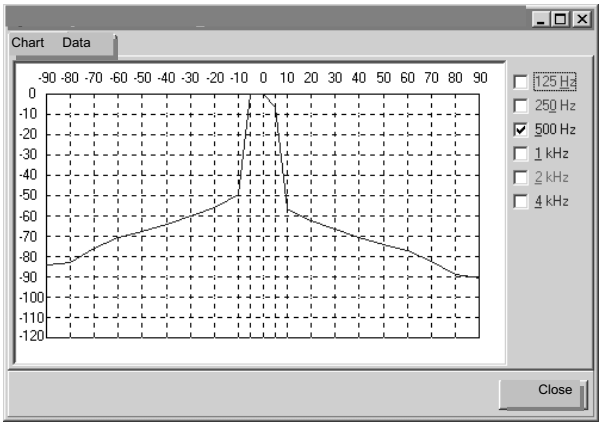


Fig. 5.34. Desired directional characteristic (the same for all frequencies). X axis represents angle, Y axis represents attenuation in dB

The effectiveness of this algorithm and the resultant speech intelligibility will depend on the proper decision made by the neural network, so the learning procedure is very important. This decision is made basing on the values of some parameters of sound that are similar to those used by the human auditory system. These parameters represent both interaural intensity ratio and interaural time difference. The first parameter, which expresses the interaural spectral magnitude ratio, is described by the following expression:

$$M^i = \frac{\min\left(\left|L^i\right|,\left|R^i\right|\right)}{\max\left(\left|L^i\right|,\left|R^i\right|\right)} \tag{5.40}$$

where:

- i – spectral component index,
- L^i, R^i – left and right signals for the ith spectral component,
- M^i – magnitude ratio for the ith spectral component

The second parameter, which expresses the interaural phase difference is described by the following expression:

$$A^i = \left|\angle L^i - \angle R^i\right| \tag{5.41}$$

where:

- $\angle$ – denotes the signal phase,
- A^i – phase difference of the ith frequency component of left and right channels

The third parameter used in learning phase is defined as:

$$D^i = \frac{\left|L^i - R^i\right|}{\left|L^i\right| + \left|R^i\right|} \tag{5.42}$$

where: D^i – relative ratio of the ith spectral component for the left and for the right channel.

It can be shown that parameters described by Eqs. (5.40) and (5.42) are in a simple functional relationship and therefore one of them is superfluous. In such a case, parameters representing a single spectral bin can consists of parameters given by Eqs. (5.40) and (5.41). In addition considering

that the above given parameters concern pairs of channels Ch_i^k and Ch_j^k, these parameters for the kth spectral bin can be rewritten as follows:

$$M_{ij}^k = \frac{\min\left(\left|Ch_i^k\right|,\left|Ch_j^k\right|\right)}{\max\left(\left|Ch_i^k\right|,\left|Ch_j^k\right|\right)} \tag{5.43}$$

$$A_{ij}^k = \angle Ch_i^k - \angle Ch_j^k \tag{5.44}$$

Thus in the second type of feature vectors 8-channel signals were examined, and hence the following sets of parameters can be considered:

- all mutual combinations of channels that yield 56 parameters per bin (***A***).
- a combination of opposite channels, which yields 8 parameters per bin (***B***).

5.4.3 System Description

The user interface of the program, prepared in Multimedia Systems Department for the purpose of training neural networks, is presented in Fig. 5.35. During the learning phase the *Mean Square Error* (MSE) was observed. MSE represents the squared error between the current value at the output of the network o and the desired response of the network d. An example of convergence of the learning process is shown in Fig. 5.36.

$$MSE = \frac{1}{PK}\sum_{l=1}^{P}\sum_{k=1}^{K}\left(o_{lk} - d_{lk}\right)^2 \tag{5.45}$$

where P is the number of training patterns, and K denotes the number of outputs

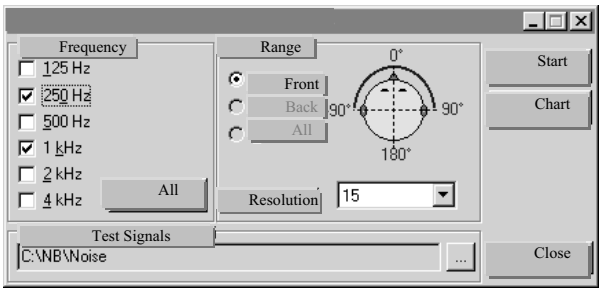


Fig. 5.35. GUI of the program used in the learning phase

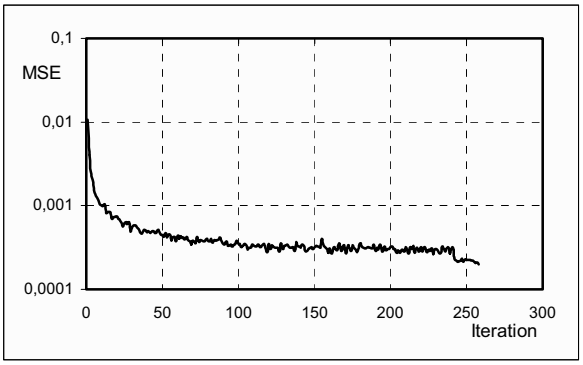


Fig. 5.36. Convergence of the learning process

Neural Network Structures and Properties

Feed-forward ANNs

As mentioned before two types of ANNs were used in experiments. First, assumptions of experiments employing standard feed-forward ANNs will be described. The proposed neural network structure and its properties were such as follows: one hidden layer consisted of 9 neurons, the standard error backpropagation algorithm with momentum was used in the learning phase. The BP learning algorithm parameters were as follows: η = 0.5 (learning rate); α = 0.01(momentum ratio). Spectral components were obtained with 512 point FFT procedure using Blackmann window with an overlap of 256 samples. The training file consists of logatoms of every 15^0 elevation. Each direction was represented by 10 sound examples (5 female and 5 male voices). In addition sounds from ±5° were used in this phase. These directions were treated similarly to 0° direction, thus the gain factor was equal to 1, whereas for other directions a value of 0.01 was used.

RNNs

Since the feed-forward neural networks do not offer such feasibility as recurrent ones – especially in the field of time series modeling or mapping of the complex process dynamics – thus also RNNs were used in experiments. The purpose of such experiments was to check whether encoding spatial information would be satisfactory based on temporal cues only.

The desired level attenuation was defined for each pattern. All signals coming from other than a front direction were treated as unwanted ones. For purposes of the experiments, eight-second excerpts were prepared representing the sound directivity of -45 ^{o} to +45 ^{o} with the step of 15 ^{o}. Since

these parameters are to be fed to a neural network, they are grouped into input vectors. The following three types of such vectors can be considered:

*type **V*1**: all spectral bins are included in a vector.

*type **V*2**: an input vector consists of parameters for a single bin and the additional information on the bin frequency.

*type **V*3**: an input vector consists only of parameters for a single bin. In this case, a neural network assumes a structure of a modular network where a separate neural subnet is dedicated for each spectral bin. The final neural decision is made on the basis of maximum outputs of all subnetworks.

The above shown division of feature vectors evokes some problems related to the size of an input vector that in turn results in the size of weight matrices of a neural network and in addition to both capacity of a neural net and selection of its architecture. Taking all the above into account only some combinations of earlier defined sets of parameters were chosen for experiments, namely: vector type ***V*3**, parameters type ***A***, sizes of an analysis frame ($N = 1024$).

In experiments both Fahlman's general and simplified algorithms (QuickPROP) (Fahlman 1998) and the **R**esilient **PROP**agation (RPROP) (Riedmiller and Braun 1993) were employed. These algorithms were reviewed in Section 3.3.4, thus only the main principles of these algorithms will be presented here.

- **Fahlman's algorithm (Fahlman I)**

The weight update rule for a single weight w_{ij} in the kth cycle is computed according to Eq. (3.111), and there is an assumption that the learning rate η^k and the momentum ratio α_{ij}^k vary according to Eqs. (3.112) and (3.113). In the formulae cited above, the constant values of the training parameters assume: $0.01 \le \eta_0 \le 0.6$, $\alpha_{\max} = 1.75$.

- **Fahlman's simplified algorithm (Fahlman II)**

In the simplified version of the QuickPROP algorithm, the weight update rule is expressed by the following relationship:

$$\Delta w_{ij}^k = \begin{cases} \alpha_{ij}^k \cdot \Delta w_{ij}^{k-1} & ;\text{for} \quad \Delta w_{ij}^{k-1} \neq 0 \\ -\eta_0 \cdot \nabla E\left(\Delta w_{ij}^k\right) & ;\text{otherwise, i.e.:} \quad \Delta w_{ij}^{k-1} = 0 \end{cases} \tag{5.46}$$

where the momentum ratio α_{ij}^k changes according to the expression below:

$$\alpha_{ij}^{k} = \min\left\{ \frac{\nabla E\left(\Delta w_{ij}^{k}\right)}{\nabla E\left(\Delta w_{ij}^{k-1}\right) - \nabla E\left(\Delta w_{ij}^{k}\right)}, \quad \alpha_{\max} \right\} \tag{5.47}$$

and the constant values of the training parameters assume are the same as in the general QuickPROP, i.e.: $0.01 \le \eta_0 \le 0.6$, $\alpha_{\max} = 1.75$.

- **RPROP algorithm**

In the case of the RPROP algorithm, the weight update rule is given by the formula based on the *signum* function (see Section 3.3.4, Eq. (3.114), where the learning rate η^k assumes values according to the rules given in Eq. (3.115). The constant values are set as follows:

$$\eta_{\min} = 10^{-6}, \ \eta_{\max} = 50 \tag{5.48}$$

$$\mu^{-} = 0.5 \ \mu^{+} = 1.2 \ 0 < \mu^{-} < 1 < \mu^{+}$$

5.4.4 Test and Results

In the testing phase various combinations of signals were introduced to the neural network inputs. Namely such signals as: tones, tone plus noise, a phoneme (logatom) plus tone, a phoneme plus noise, phonemes and phrases were employed as testing material. Always one of the signals was coming from the front direction (0^0), and the other was the unwanted one and was localized at the angle between 15^0 to 90^0 (horizontal plane).

First, results obtained for the feed-forward neural network-based beamformer will be now shown. An example of spatial characteristics obtained after the learning phase are presented in Fig. 5.37. As expected sharper minima and maxima were obtained for higher frequency spatial characteristics for the whole angle range. The slope of low frequency characteristics for 15^0-90^0 azimuth is very smooth.

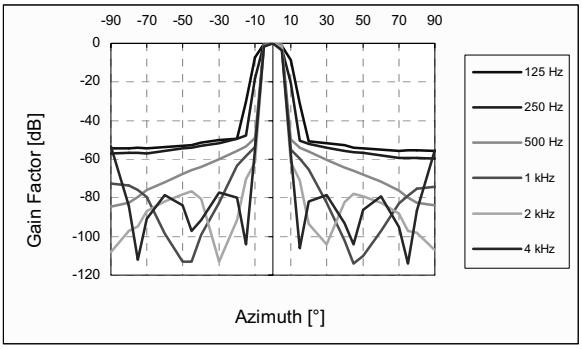
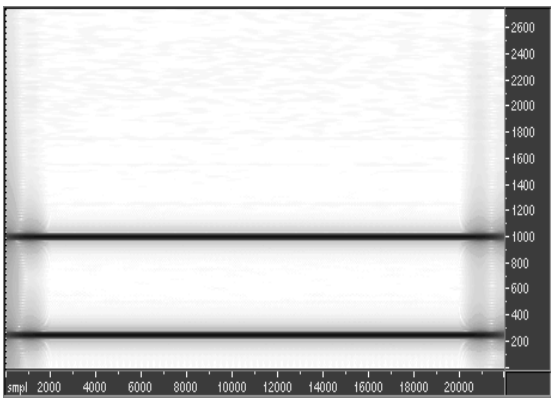


Fig. 5.37. Spatial characteristics of the ANN-based filtration algorithm obtained with a multi-tone signal

In Figs. 5.38-5.4 the examples of a signal spectral representation (sonograms) before and after processing are shown. In Fig. 5.38 the target signal is 1kHz and the concurrent one is 250 Hz. On the other hand, in Fig. 5.39 the disturbing signal is filtered noise of the center frequency equals to 250 Hz. As is seen from Fig. 5.38 and Fig. 5.39 disturbing signals are strongly attenuated, but the proposed algorithm causes some distortions that are noticeable in the spectral domain. Especially interesting is the processing result shown in Fig. 5.39. In this case noise around 500Hz and above this frequency appears. However the signal-to-noise ratio equals to -60dB. In addition the disturbing signal is strongly attenuated, so the distortions do not influence substantially the overall quality of audio.

a. before processing



(Legend to Fig. 5.38, see next page)

b. after processing

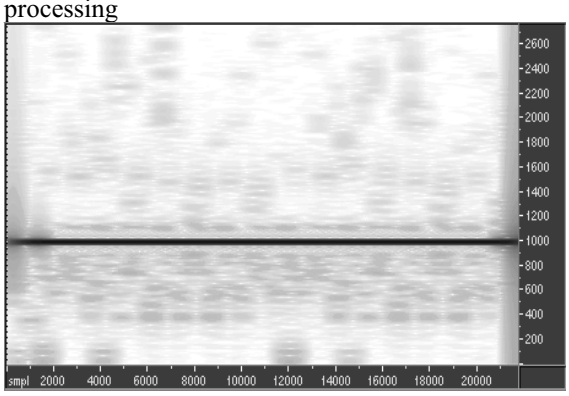
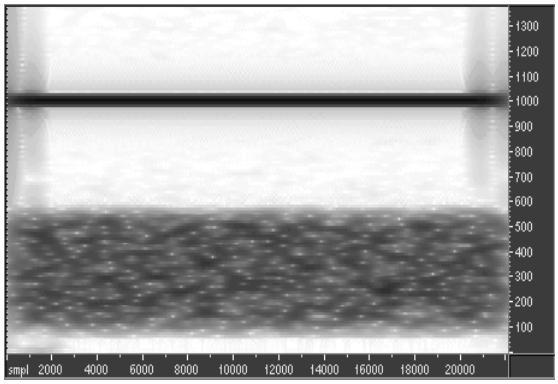


Fig. 5.38. Spectral representation of signals (1kHz, 0°)+(250Hz, azimuth 45°)

a.



b.

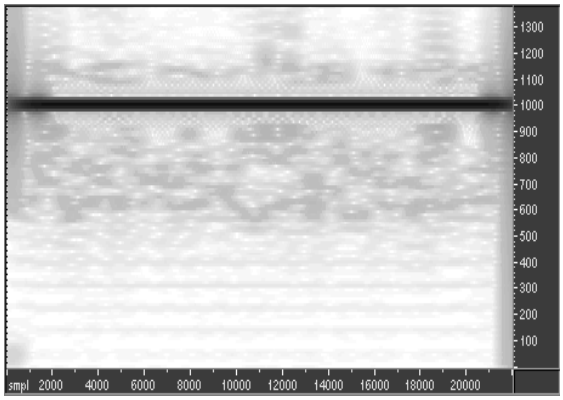
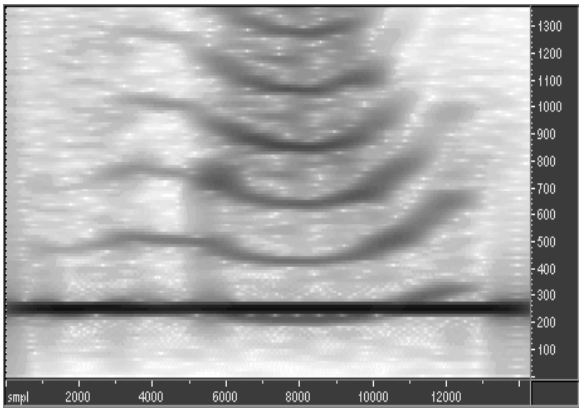


Fig. 5.39. Spectral representation of signals (1kHz, 0°)+(filtered noise, f_0=250Hz, azimuth 45°), before processing (a), after processing (b)

In Fig. 5.40 another combination of signals that was processed by the neural beamformer is shown. In this case the target signal was a logatom and the disturbing one was a 250 Hz harmonic tone. As is seen from the sonogram analysis the target signal has got a formant around the same frequency as such of the concurrent signal. That is why the algorithm after processing cuts off this frequency along with the formant.

In Fig. 5.41 time-domain presentation of neural network filtration effects is shown.

a.



b.

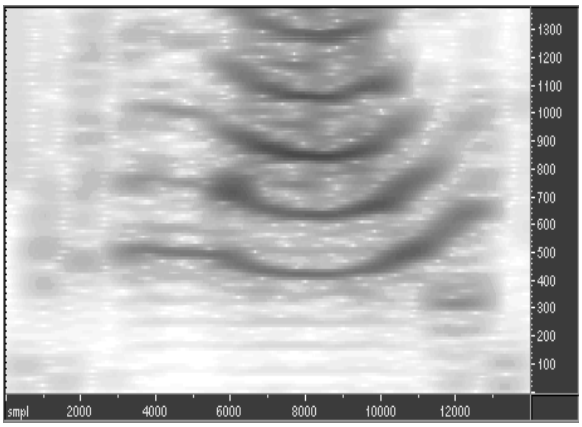


Fig. 5.40. Spectral representation of signals (phoneme, 0°)+(signal f_0=250Hz, azimuth 45°), before processing (a), after processing (b)

After processing various combinations of signals and azimuths it was observed that worse filtration effects were observed when a concurrent

signal was close to the target signal (15^0 azimuth). In this case the dependence of the filtration effects on the character of the signal was also noticed. It can be also observed that definitions of parameters (Eq. 5.40) and (Eq. 5.41) cause that signals of the same spectrum composition coming from concurrent directions may not be effectively filtered out by such a beamformer algorithm. This is the most important drawback of the proposed method of spatial filtering, however in such a case a classical beamformer does not perform well, either.

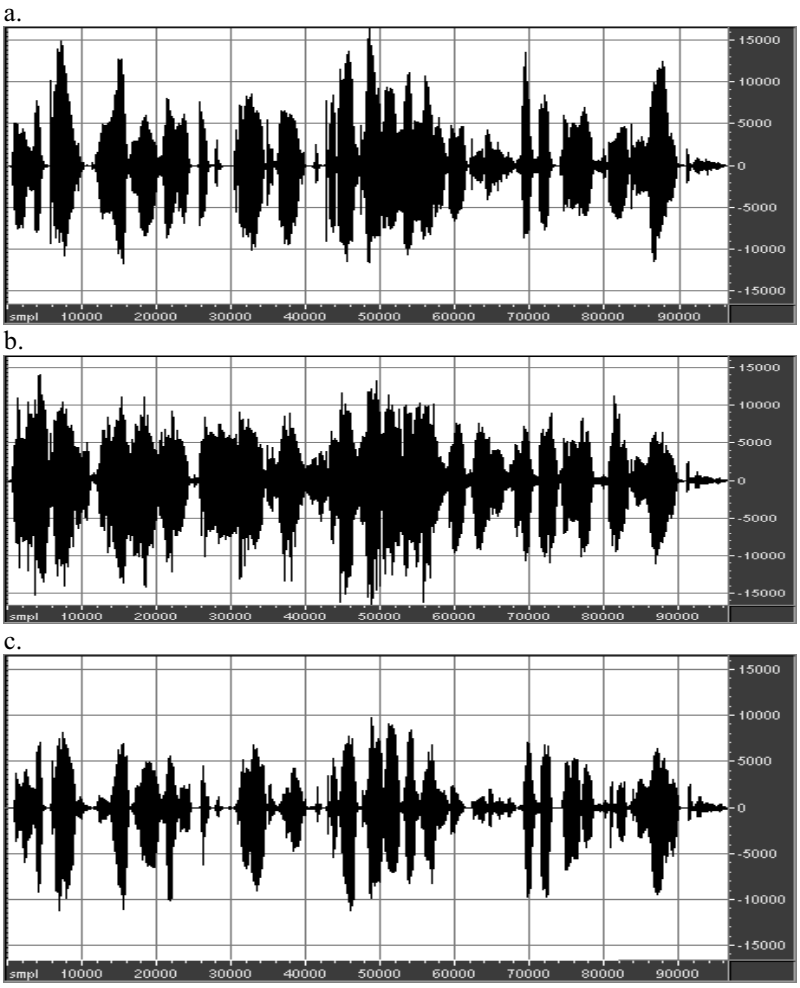


Fig. 5.41. Filtration of two sentences (time domain representation); original signal before processing (a); signal mixture – original signal + disturbing signal from 45° (b); result of NN-based filtration (c)

Examples of the results obtained by the RNN-based beamformer are shown in Table 5.7. There were 515 training vectors and 221 testing vectors. As seen from table these results depend on both the training algorithm and the signal direction. The best scores were obtained for the azimuth of 45^0. On the other hand, no conclusions as to the best training algorithm could be derived on the basis of the obtained results.

Table 5.7. Results of direction detection for: *vector type* - ***V*3**, N = 1024, parameter type - ***A***, training/testing vectors = 515/221

Direction	Fahlman I [%]	Fahlman II [%]	RPROP [%]
-45°	90	92	89
-30°	89	87	88
-15°	88	89	90
0°	90	90	88
15°	86	82	85
30°	85	85	84
45°	87	88	88

Comparison with a standard adaptive beamformer

In order to compare results of the elaborated system with the classical algorithm a well-known Griffith-Jim adaptive beamformer (1982) working in the time domain was used in investigations. It consisted of 64 taps. In all tests a male speech was used as a desired signal. The interfering signals were as follows: male speech, harmonic tones and white noise, all presented at 45^0 position. Results are gathered in Tab. 5.8.

Table 5.8. Comparison of the ANN spatial filter with the Griffith-Jim beamformer

Noise	Griffith-Jim beamformer	ANN spatial filter
male speech 45^0	good noise reduction, almost no imperfections	good noise reduction, almost no imperfections
harmonic tones 45^0	whole noise reduction, distortion observed	whole noise reduction, small imperfections
white noise 45^0	little noise reduction, small imperfections	big noise reduction, small imperfections

The obtained results demonstrate that a non-linear filter based on neural network provides an effective tool for the detection of sound source localization. It was additionally shown that both standard feed-forward ANNs and recurrent neural networks could be used for the purpose of sound localization. It should be remembered here that these beamformes used different approach to feature extraction. Neural networks-based beamformers

can cause a significant increase in the signal-to-noise ratio. Such results open a possibility to employ the neural network-based sound localization algorithms to experimental teleconference systems.

References

McAdams S, Winsberg S (1999) Multidimensional scaling of musical timbre constrained by physical parameters. J Acoust Soc Am 105:1273

Adkins CN, Turtora JJ (1996) A Broadband Beam-Former with Pole – Zero Unconstrained Jammer Rejection in Linear Arrays. IEEE Trans SP 44 (2): 438 – 441

Allen JB, Hall JL, Jeng PS (1990) Loudness growth in ½ octave bands (LGOB) – A procedure for the assessment of loudness. J Acoust Soc Am 88, No 2

Aoki S, Okamoto M (1999) Audio teleconferencing system with sound localization effect. In: Proc 137th Acoust Soc Am Meeting, Berlin, Germany

Beauchamp JW (1993) Unix Workstation Software for Analysis, Graphics, Modification, and Synthesis of Musical Sounds. In: 94th Audio Eng Soc Conv. Berlin, Preprint No 3479

Berdugo B, Doron MA, Rosenhouse J, Azhari H (1999) On Direction Finding of an Emitting Source from Time Delays. J Acoustical Soc of Am 106: 3355–3363

Bodden M (1993) Modeling Human Sound-Source Localization and the Cocktail-Party-Effect. Acta Acustica 1: 43–55

Bosc P, Kacprzyk J (eds) (1995) Fuzziness in Database Management Systems. Physica-Verlag (Springer-Verlag), Heidelberg

Bose BK (1994) Expert System. Fuzzy Logic and Neural Network Applications in Power Electronics and Motion Control. IEEE 82, 8: 1303-1323

Brandstein MS (1997) A Pitch-Based Approach to Time-Delay Estimation of Reverberant Speech. In: Proc of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, Mohonk, New Paltz, NY, USA

De Bruijn A (1978) Timbre-Classification of Complex Tones. Acustica 40:108-114

Caddy S, Pollard HF (1957) Transient Sounds in Organ Pipes. Acustica 7:227-280

Chern S-J, Lin S-H (1994) An Adaptive Time Delay Estimation with Direct Computation Formula. J of Acoustical Soc of Am 96: 811–820

Cook PR (1999) Music, Cognition, and Computerized Sound, An Introduction to Psychoacoustics. MIT Press, Cambridge, Massachusetts, London, England.

Cosi P, De Poli G, Parnadoni P (1994) Timbre characterization with Mel-Cepstrum and Neural Nets. In: Proc of the ICMC'94, pp 42-45

Czyzewski A (1998) Speaker-Independent Recognition of Isolated Words Using Rough Sets. J Information Sciences 104: 3-14

Czyzewski A (2003a) Intelligent Acquisition of Audio Signal Employing Neural Network and Rough Set Algorithms. Rough-Neuro Computing: A Way To

Computing With Words, Springer Verlag, Series on Artificial Intelligence, 521 - 541

Czyzewski A (2003b) Automatic Identification of Sound Source Position Employing Neural Networks and Rough Sets. Pattern Recognition Letters 24: 921 - 933

Czyzewski A, Kostek B (2002) Expert Media Approach to Hearing Aids Fitting; Int Journ of Intelligent Systems 17: 277 - 294

Czyzewski A, Kostek B, Krolikowski R (2001) Neural Networks Applied to Sound Source Localization. In: Proc 110th Audio Eng Soc Conv, Amsterdam

Czyzewski A, Kostek B, Lasecki J (1998) Microphone Array for Improving Speech Intelligibility. In: Proc 20 Tonmeistertagung, International Convention on Sound Design, Stadthalle Karlsruhe, Germany, 428-434

Czyzewski A, Kostek B, Skarzynski H (2002a) Applications of Computer Technology to Audiology and Speech Theraphy, EXIT Academic Press (*in Polish*)

Czyzewski A, Kostek B, Suchomski P (2000) Expert System for Hearing Aids Fitting. In: Proc 108th Audio Eng Soc Conv, Preprint No 5094, Paris, France

Czyzewski A, Krolikowski R (2001a) Acquisition of Acoustic Signals Assisted by Recurrent Neural Networks. In: Proc 17th International Congress on Acoustics

Czyzewski A, Krolikowski R (2001b) Automatic Identification of Sound Source Direction Based on Neural Networks. In: Proc 142nd Meeting of the Acoustical Society of America, J Acoust Soc Am, 4aSP9 No 5 vol 110 p 2741, Fort Lauderdale, USA

Czyzewski A, Lasecki J (1999) Neural network-based spatial filtration algorithm for a 2-D microphone array. Acta Acustica - Proc of the Forum Acusticum '99, Berlin, March 14-19. Abstract book, Hirzel Verlag, 4aSP7, p 326

Czyzewski A, Lasecki J, Kostek B (1999) Computational Approach to Spatial Filtering. In: Proc 7th European Congress on Intelligent Techniques and Soft Computing, (EUFIT'99), Aachen, Germany, p 242

Czyzewski A, Skarzynski H, Kostek B, Kochanek K, Sliwa L (2002b) New diagnostic tools in the family doctor practice. In: Proc International Conference on TELEMEDICINE inter- and intradisciplinary application, May 23-25, Jablonna, Poland

Dubois D, Prade H (1999) Fuzzy Sets in Approximate Reasoning, Part 1: Inference with Possibility Distributions. In: Fuzzy Sets and Systems, supplement to vol 100 (A selection of the most cited papers in Fuzzy Sets and Systems), pp 73-132

Dubois D, Prade H, Ughetto L (2002) A New Perspective on Reasoning with Fuzzy Rules. In: Proc of the 2002 AFSS International Conference on Fuzzy Systems, Calcutta: Advances in Soft Computing, 1-11

Fahlman S (1988) An Empirical Study of Learning Speed in Back-Propagation Networks, Technical Report CMU-CS-88-162 of Carnegie Mellon University in Pittsburgh, USA

Frost OL (1972) Adaptive least-squares optimization subject to linear equality constraints. Stanford Univ

Fuller R (1999) On fuzzy reasoning schemes. In: Carlsson C (ed) The State of the Art of Information Systems in 2007, TUCS General Publications, No 16, Turku Centre for Computer Science, Abo, pp 85-112, URL: http://www.abo.fi/~rfuller/pgs97.pdf

Grey JM (1977) Multidimensional perceptual scaling of musical timbres. J Acoust Soc Am 61:1270-1277

Griffiths LJ, Jim CW (1982) An alternative approach to linearly constrained adaptive beamforming. IEEE Trans Antennas and Propagation 30: 27-34

Hartmann WM (1999) How We Localize Sound. Physics Today 11: 24–29

Herrera P, Amatriain X, Battle E, Serra X (2000) Towards Instrument Segmentation for Music Content Description: A Critical Review of instrument classification techniques. In: Proc Int Symp on Music Information Retrieval, ISMIR'2000, Plymouth

Hua L, Yuandong J (1994) Fuzzy-Logic Tools on Tap for IC Wafers. IEEE Circuits and Devices, pp 30-35

Iverson P, Krumhansl CL (1993) Isolating the dynamic attributes of musical timbre. J Acoust Soc Am 94

Jacovitti G, Scarano G (1993) Discrete time techniques for time delay estimation. IEEE Trans Signal Process 41, 525–533

Jensen K (2001) The timbre model, Workshop on current research directions in computer music, Barcelona

Kacprzyk J, Feddrizzi M (eds) (1992) Fuzzy Regression Analysis 1. Omnitech Press, Warsaw and Physica-Verlag (Springer-Verlag), Heidelberg New York

Khalil F, Lullien JP, Gilloire A (1994) Microphone Array for Sound Pickup in Teleconference Systems. J of Audio Eng Soc 42: 691–700

Kosicki R (2000) Examination of Spatial Filter Characteristics based on Neural Networks. MSc Thesis, Multimedia Systems Department, Gdansk University of Technology (*in Polish*), Kostek B (supervisor)

Kosko B (1997) Fuzzy Engineering. Prentice-Hall Intern Ed, New Yersey

Kosko B (1992) Neural Networks and Fuzzy Systems. Prentice-Hall, London

Kostek B (1992) Untersuchungen an Orgeltrakturen unter dem Aspekt musikalischer Artikulierung. Inl Teil A Fortschritte Der Akustik, Proc DAGA '92, Berlin, pp 245-248

Kostek B (1994a) Application des réseaux de neurones pour l'analyse de l'articulation musicale, J de Physique IV 4:597-600

Kostek B (1994b) Intelligent Control System Implementation to the Pipe Organ Instrument. In: Ziarko WP (ed) Rough Sets, Fuzzy Sets and Knowledge Discovery. Springer-Verlag, London, pp 450-457

Kostek B (1997) Articulation-Related Features in the Pipe Organ Sound. Archives of Acoustics 22:219-244

Kostek B (1999) Soft Computing in Acoustics, Applications of Neural Networks, Fuzzy Logic and Rough Sets to Musical Acoustics, Studies in Fuzziness and Soft Computing. Physica Verlag, Heidelberg New York

Kostek B (2003) "Computing with words" Concept Applied to Musical Information Retrieval. Electronic Notes in Theoretical Computer Science 82, No 4

Kostek B, Czyzewski A (1991) Articulation Features in the Digitally Controlled Pipe Organ. J Audio Eng Soc 39:382, 90th AES Convention, Preprint No 3023, Paris

Kostek B, Czyzewski A (1992) Computer Modelling of the Pipe Organ Valve Action. J Audio Eng Soc 40:440, 92nd AES Convention, Preprint No 3266, Vienna

Kostek B, Czyzewski A (1993) Investigation of Articulation Features in Organ Pipe Sound. Archives of Acoustics 18:417-434

Kostek B, Czyzewski A (2001a) Representing Musical Instrument Sounds for Their Automatic Classification. In: J Audio Eng Soc vol 49:768-785

Kostek B, Czyzewski A (2001b) Multimedia Techniques Applied to Health Care Procedures - Hearing Aid Fitting Expert System, 46 Internationales Wissenschaftliches Kolloquium, Ilmenau, Germany

Kostek B, Czyzewski A (2001c) A method for the automatic hearing aid fitting employing speech in noise. In: Proc 142nd Acoust Soc of Am Meeting 100, No 5, 2pPP10, Fort Lauderdale, USA

Kostek B, Czyzewski A, Lasecki J (1999) Spatial Filtration of Sound for Multimedia Systems. In: Proc IEEE Signal Processing Society 3rd Workshop on Multimedia Signal Processing, pp 209-213, Copenhagen, Denmark

Kostek B, Czyzewski A, Skarzynski H (2001) Internet-Based Automatic Hearing Assessment System. 46 Internationales Wissenschaftliches Kolloquium, pp 87-89, Ilmenau, Germany

Kostek B, Suchomski P, Czyzewski A (2004) A system for fast & precise hearing AIDS fitting. In: Proc 2nd International Conference on Telemedicine and Multimedia Communication, Kajetany, Abstr in Elec J of Pathology and Histology

Krimphoff J, McAdams S, and Winsberg S (1994) Caracterisation du timbre des sons complexes. II Analyses acoustiques et quantification psycho-physique. J Phys vol 4:625-628

Krolikowski R, Czyzewski A, Kostek B (2001) Localization of Sound Sources by Means of Recurrent Neural Networks. Lecture Notes in Computer Science 2005, Springer-Verlag, pp 603 – 610

Kuncheva LI, Kacprzyk J (2000) Fuzzy Classifier Design. Physica-Verlag

Larsen PM (1980) Industrial Applications of Fuzzy Logic Control. International Journal of Man-Machine Studies 12:3-10

Lasecki J, Czyzewski A (1999) Neural Network-Based Spatial Filtration Algorithm for 2-Microphone Array. In: Proc Acoust Soc Am Meeting, Berlin, Germany

Lasecki J, Czyzewski A, Kostek B (1998) Neural Network-Based Algorithm For the Improvement of Speech Inteligibility. In: Proc of 20th International Convention on Sound Design, pp 428-434, Karlsruhe

Lasecki J, Kostek B, Czyzewski A (1999) Neural Network-Based Spatial Filtration of Sound. In: Proc 106th Audio Eng Soc Convention, Preprint No 4918 (J4), Munich, Germany

Lindsay AT, Herre J (2001) MPEG-7 and MPEG-7 Audio – An Overview. J Audio Eng Soc 49:589-594

Mahieux Y, le Tourneur G, Saliou A (1996) A Microphone Array for Multimedia Workstations. J of Audio Eng Soc 44: 365–372

Mamdani EH (1977) Applications of Fuzzy Set Theory to Control Systems: A Survey. In Gupta MM, Saridis GN, Gaines BR (eds) Fuzzy Automata and Decision Processes. North-Holland, New York, pp 1-13

Mendel J (1995) Fuzzy Logic Systems for Engineering: A Tutorial. In: Proc of the IEEE 83, No 3, pp 345-377

Misdariis N, Smith BK, Pressnitzer D, Susini P, McAdams S (1998) Validation of a Multidimensional Distance Model for Perceptual Dissimilarities among Musical Timbres. In: Proc 16th International Congress on Acoustics and 135th Meeting Acoustical Society of America, Seattle, Washington

Nowosielski J (1997) Self Fitting and Self Controlled Hearing Aid. In: Proc 3rd European Congress on Audiology, Prague

Pawlak Z (1982) Rough Sets, J Computer and Information Science 11:341-356

De Poli G, Piccialli A, Roads C (eds) (1991) Representations of Musical Signals. MIT Press, Cambridge

De Poli G, Canazza S, Drioli C, Roda A, Vidolin A, Zanon P (2001) Analysis and modeling of expressive intentions in music performance. In: Proc of the Intern Workshop on Human Supervision and Control in Engineering and Music. Kassel, URL: http://www.engineeringandmusic.de/workshop

Pratt RL, Doak PE (1976) A subjective rating scale for timbre. J Sound and Vibration 45:317-328

Pratt RL, Bowsher JM (1978) The subjective assessment of trombone quality. J Sound and Vibration 57:425-435

Reuter C (1996) Karl Erich Schumann's principles of timbre as a helpful tool in stream segregation research. In: Proc II Intern Conf on Cognitive Musicology, Belgium

Riedmiller M, Braun H (1993) A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. In: Proc IEEE International Conf on Neural Networks, 586-591, San Francisco

Skarzynski H, Czyzewski A, Kostek B (2002) Principles and acoustical foundations of the computer-based hearing screening method. In: Proc 144th Meeting of the Acoust Soc of Am (First Pan-American/Iberian Meeting on Acoustics), J Acoust Soc Am, No 5 112, Cancun, Mexico

Skowron A (1994) Data Filtration: a Rough Set Approach. In Ziarko WP (ed) Rough Sets, Fuzzy Sets and Knowledge Discovery. Springer-Verlag, London, pp 108-118

Suchomski P (2005) Hearing Prothesis Fitting System Based on Speech Perception in Noise. PhD Thesis, Multimedia Systems Department, Gdansk University of Technology (*in Polish*), Kostek B (supervisor)

Sugeno M (1985) An Introductory Survey of Fuzzy Control. Information Sciences 36, pp 59-83

Szczepaniak PS, Segovia J, Kacprzyk J, Zadeh LA (2003) Intelligent Exploration of the Web. Physica-Verlag

Takagi T, Sugeno M (1985) Fuzzy Identification of Systems and its Application to Modelling and Control. IEEE Trans on Systems, Man and Cybernetics 15, pp 116-132

Yager RR (1992) Implementing fuzzy logic controllers using a neural network framework. Fuzzy Sets and Systems 48:53-64

Yamakawa T (1989) Stabilization of an Inverted Pendulum by a High-Speed Fuzzy Logic Controller Hardware System. Fuzzy Sets and Systems 32:161-180

Yu X, Kacprzyk J (2003) Applied Decision Support With Soft Computing. Springer-Verlag

Veen BD, Buckley KM (1988) Beamforming: A Versatile Approach to Spatial Filtering. IEEE Acoust, Speech and Sig Proc Mag 5(2): 4-24

Vercoe B, Gardner W, Schreier E (1998) Structured Audio: Creation, Transmission, and Rendering of Parametric Sound Representations. In: Proc IEEE 86: 922-940

Wang H, Chu P (1997) Voice Source Localization for Automatic Camera Pointing System in Videoconferencing. In: Proc IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, Mohonk, New Paltz, NY, USA

Widrow B, Stearns SD (1985) Adaptive Signal Processing. Prentice-Hall

Wessel DL (1979) Timbre Space as a Musical Control Structure. J Computer Music 3:45-52

Zadeh L (1965) Fuzzy Sets. Information and Control 8:338-353

Zadeh L (1994) Fuzzy logic, neural networks, and soft computing. Communications of the ACM, 37: 77-84

Zadeh L (1996) Fuzzy logic = Computing with words. IEEE Trans on Fuzzy Systems 2: 103-111

Zadeh L (1999a) Fuzzy Sets as a Basis for a Theory of Possibility. In: Fuzzy Sets and Systems, supplement to vol 100 (A selection of the most cited papers in Fuzzy Sets and Systems), pp 9-34

Zadeh L (1999b) From Computing with Numbers to Computing with Words - From Manipulation of Measurements to Manipulation of Perceptions. IEEE Trans on Circuits and Systems 45: 105-119

Zadeh L, Kacprzyk J (eds) (1992) Fuzzy Logic for the Management of Uncertainty. Wiley, New York

Zemankowa M, Kacprzyk J (guest eds) (1993) Integrating Artificial Intelligence and Databases Technologies. Special Issue of J of Intelligent Information Systems 2, No 4

URL: http://www.soft-computing.de/def.html

URL: http://www.pa.info.mie-u.ac.jp/WFSC/

6 SYNESTHETIC ANALYSIS OF AUDIO-VISUAL DATA

6.1 Data Acquisition

6.1.1 Problem Overview

Interaction between two perceptual modalities, seeing and hearing, their interaction and mutual reinforcement in a complex relationship was a subject of many research studies. The term synesthesia, under which the phenomenon is known, is related to involuntary joining in which the real information of one sense is accompanied by a perception in another sense. Attempts to assess the degree of this interaction have for many reasons been of special interest.

Contemporary digital video, film or multimedia presentations are often accompanied by the surround sound. Techniques and standards involved in digital video processing are much more developed than concepts underlying the process of recording and mixing a multichannel sound. The main challenge in sound processing in a multichannel system is to create an appropriate basis for the related multimodal context of visual and sound domains. Therefore, one of the purposes of experiments is to study in which way and how the surround sound interferes or is associated with the visual context. This kind of study was hitherto carried out when a two-channel sound technique was associated with a stereo TV. However, there is not much study done yet that associates real correlation between the surround sound and digital video presented at the TV screen. The main issue in such experiments is the analysis of how visual cues influence the perception of the surround sound. This problem will be solved by applying fuzzy logic to process subjective test results.

There are many scientific reports showing that human perception of sound is affected by image and vice versa. For example, Stratton in his ex-

Bożena Kostek: *Perception-Based Data Processing in Acoustics*, Studies in Computational Intelligence (SCI) **3**, 389–417 (2005)
www.springerlink.com

periments carried out at the end of 19th century proved that visual cues can influence directional perception of sound. This conclusion was confirmed later by others. Gardner experimentally demonstrated how image can affect the perceived distance between the sound source and the listener (Gardner 1968). The phenomenon of interference between the sound and vision stimuli was reported also by Thomas (Thomas 1941), Witkin, Wapner and Leventhal (1952). Very important experiments demonstrating interaction between audio and video in stereo TV were made by Brook et al. (1984), Gardner (1968), Wladyka (1987). Such experiments were also carried out by Sakamoto et al. (1981, 1982) in the context of controlling sound-image localization in stereophonic reproduction. Komiyama (1989) performed subjective evaluation of angular displacement between picture and sound directions for HDTV sound systems. Some years later, Bech, Hansen and Woszczyk (1995) and also Woszczyk et al. (1995) tried to assess audio-visual interaction for Home Theater Systems. They examined in which way two perceptual modalities, seeing and hearing, interact and reinforce each other in a complex relationship. Effects of the investigations were shown during Audio Engineering Conventions, and concerned the experimental results for the subjective attribute, namely space. The factors investigated were: basic stereo width, sub-woofer, surround sound concept, screen size, etc. Following this research, other authors tried to answer some questions concerning audio-visual correlation. For example, Beerends and de Caluwe (1999) discussed how video quality influences the perceived audio quality of an audiovisual stream, and in addition they considered in which way audio and video quality contribute to the overall perceived audiovisual quality. The main conclusion in their experiments was that video quality contributes significantly to the subjectively perceived audio quality (Beerends and de Caluwe 1999). Also, Hollier and Voelcker (1997) investigated video quality influence on audio perception. Later, Dvorko and her co-author discussed the results of theoretical and experimental researches of psychophysical and aesthetic aspects of sound and picture interaction (Dvorko and Ershov 1998). Bruijn and Boone (2000) carried out subjective tests to investigate the effects in the context of a life-size video conferencing system. Their investigation showed that the non-identical perspectives of the audio and video reproductions had a significant influence on subjects' evaluation of the total system. They proposed solutions to improve the matching of the audio and video scenes for a large listening area. The team from the Multimedia Systems Department started their investigations on audio-visual correlation in 2000, and many research reports have appeared since that time in the form of conference papers, journal articles (Czyzewski et al 2000a, 2000b, 2001; Kostek 2003)

and students' M.Sc. theses (Florek and Szczuko 2002; Kaminski and Malasiewicz 2001; Smolinski and Tchorzewski 2001).

The subject of audio-visual correlation was also pursued by Zielinski et al. (2003). These experiments focused on the standard 5.1 multichannel audio set-up according to the ITU recommendation and were limited to the optimum listening position. The obtained results of the formal listening test show that in general listeners prefer the limitation of channels to the limitation of bandwidth, for a given 'information rate'. However, for some program material with foreground content (direct sound) in the rear channels, limitation of either parameter has a similar effect.

The investigation by Rumsey et al. (2004) aimed to discover the effect of involvement in an interactive task on the perception of audio-visual asynchrony in a computer game environment. An experimental game was designed to test the investigated phenomenon. The experiment tested only audio lag conditions. It was found that within the confines of the experimental method, the threshold of perception was increased in the interactive game condition by approximately 40ms (±20ms), which is a small but statistically significant value.

However, there still is no clear answer to the question how the video influences the localization of virtual sound sources in multichannel surround systems (e.g. DTS). Therefore, there is a need of systematic research in this area, especially as sound and video engineers seek such information in order to optimize the surround sound. The results of this kind of research may improve production of movie soundtracks, recording of music events and live transmissions, thus the resulting surround sound may seem more natural to the listener. The experiments are based on subjective testing of a group of people, so-called experts, listening to the sound with- and without vision. The obtained results are processed in order to find some hidden relations underlying the influence of video on the perception of audio, particularly with regard to the influence of video to the directivity of localization of sound sources in the surrounding acoustical space. Some soft computing methods could be used to process subjective test results, bringing better results of the analysis than statistical methods, particularly if the number of tests and involved experts are reasonably small. An approach to such an application is presented in the following paragraphs. The proposed method of analysis of subjective opinion scores could be also used in other domains than audio-video perception investigation (public opinion analysis etc.).

6.1.2 Subjective Test Principles

Results of such experiments may show in which cases and in what way the video can affect the localization of virtual sound sources. In most cases video 'attracts' the attention of the listener who in consequence localizes the sound closer to the screen center. Therefore, this effect can be called the audio-visual proximity effect.

In the experiments, two rooms were used: an auditory room and a control room, which are acoustically separated. Video was projected from the control room to the auditory room through the window between these two rooms. The place for a listener was positioned in a so-called 'sweet-spot' (see Fig. 6.1). Denotations in Fig. 6.1 refer to channels of the 5.1 sound system (L - left, C - center, R - right, LS - left surround, RS - right surround). Points in this figure present phantom sound source configuration used in localization tests.

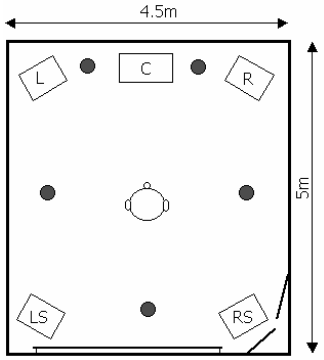


Fig. 6.1. Auditory room

During tests AC-3 (Dolby Digital) audio encoded and MPEG2 video encoded files were used. Sound files were prepared in the Samplitude 2496 application and then exported to the AC-3 encoder. The following equipment was used during the tests: a computer with a built-in DVD player, an amplifier with a Dolby Digital decoder, a video projector, a screen (dimensions: 3x2 m), loudspeakers.

Calibration Procedure

First, the calibration procedure was performed, during which the levels of received signals were checked in all loudspeakers of the 5.1 system. In Table 6.1 sound level measurement results are shown for various placement of the sound level meter. As seen from the table, the only adjustment to be done was due to the difference of level between the left (L) and right (R)

channels. Other differences in level measurement were negligible or appeared due to sound system requirements.

Table 6.1. Sound level measurement results for all channels in the 5.1 system in various configurations

Channel/level [dBA]				
L	C	R	LS	RS
All loudspeakers placed 1 m from sound level meter				
86.2	85.8	86.4	83.8	83.8
All loudspeakers placed as in the Home Theater system (distance from the sound level meter equal to 1 m				
85.4	85.8	86.6	82.6	82.8
All loudspeakers placed as in the Home Theater system (sound level meter placed in sweet-spot)				
82.4	81.8	83.0	80.2	80.8

In addition, the configuration in which the sound level meter was placed in the sweet-spot was used again. This time the reference signal was attenuated to –3, and –6 dB, respectively. Results of this part of the calibration procedure are contained in Table 6.2.

The last part of the calibration procedure was related to a proper localization of directions (see Fig. 6.1). Several students from the Multimedia Systems Department participated in tests. It was found that sound is properly localized in the system.

Table 6.2. Sound level measurement results for all channels in the 5.1 system (reference signal attenuated to –3, and –6 dB; sound level meter placed in sweet-spot)

Channel/level [dBA]				
L	C	P	LS	PS
76.0	74.4	76.2	73.0	74.2
69.6	69.0	69.8	67.4	68.2

Preliminary Listening Tests

In the preliminary experiments the arrangement of loudspeakers was as follows: four loudspeakers were aligned along the left-hand side of the screen (Fig. 6.2). In this case, the first loudspeaker was placed at the edge of the room, whereas the fourth one was positioned under the screen. This arrangement allowed for showing how the visual object can affect the angle of the subjectively perceived sound source.

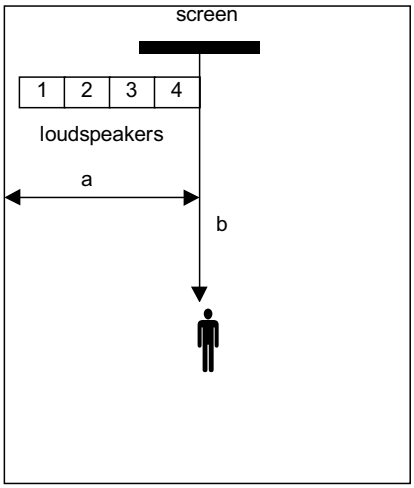


Fig. 6.2. Arrangement of loudspeakers during the tests

The experiment scenario was as follows. In the first phase of the experiment white noise was presented from the loudspeakers in random order. The expert's task was to determine from which loudspeaker the sound was heard. Then, in the second phase a blinking object was displayed in the center of the screen with a synchronously generated white noise. In the center of a circle a one-digit number was displayed. Each time the circle was displayed the number changed in order to draw the listener's attention to the picture. Obtained results show that the image proximity effect is speaker dependent, however most experts' results clearly demonstrate the mentioned effect. The most prominent data showing this effect is depicted in Fig. 6.3. The shift in the direction to the centrally located loudspeaker is clearly visible.

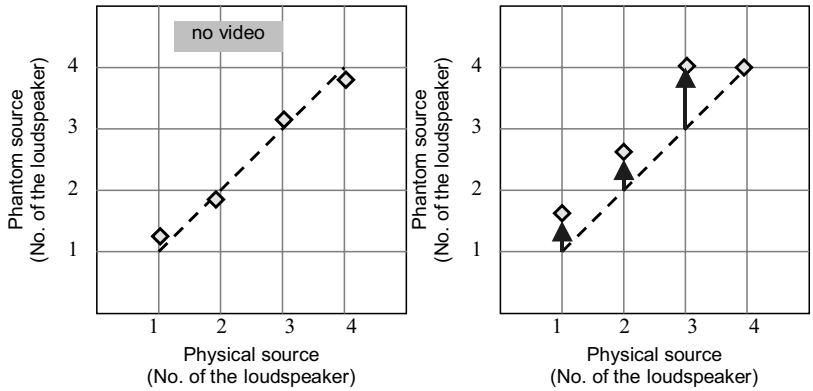


Fig. 6.3. Comparison of answers of an expert for two types of experiments: without video/with video

6.1.3 Data Processing

The subjective tests presented below aimed at finding a relation between precise surround directivity angles and semantic descriptors of the horizontal plane directions. It is hard to expect an expert to be exact in localizing phantom sources in the surround stereophonic base and to provide precise values of angles. On the other hand, it seems quite natural that an expert will localize a sound using such directional descriptors as: *left, left-front, front, right-front, right, rear-right, rear, rear-left*. Thus, first series of the experiment should consist in mapping these descriptors to angles as in Fig. 6.4.

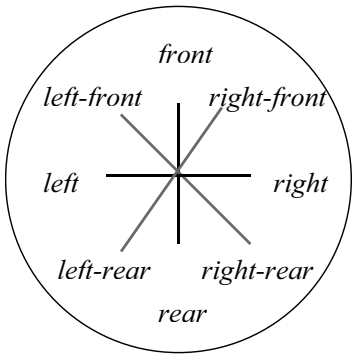


Fig. 6.4. Questionnaire form used in the first stage of experiments

In this phase of the investigations, sound samples recorded in the anechoic chamber should be presented to a group of experts. The experts, while listening to sounds excerpts, are instructed to rate their judgements of the performance using descriptions introduced above. In order to obtain statistically validated results, various sound excerpts should be presented to a sufficiently large number of experts. This procedure is based on the concept of the Fuzzy Quantization Method (FQM) applied to acoustical domain (Kostek 1999). Since the experimenter knows to what angle a given sound was assigned, thus this stage of experiments results in mapping semantic descriptors received from experts to particular angles describing the horizontal plane.

In order to simplify this phase of tests, localization sphere should be divided into 5° steps. Fig. 6.5 shows exemplary mapping of the front membership function. All other membership function should be estimated in a similar way (see Fig. 6.6).

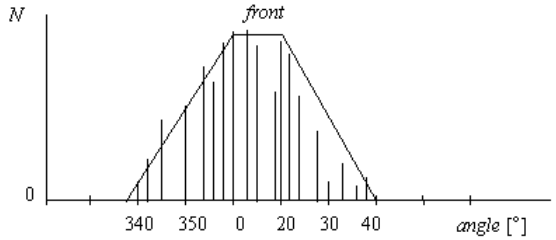


Fig. 6.5. Experts' votes for the front membership function, *N* - the number of experts voting for particular values of localization (variable: *angle*)

As shown in Figs. 6.5 and 6.6, the distribution of the observed instances may suggest a typical trapezoidal shape of a membership function. In the next step of the analysis, membership functions should be identified with the use of some statistical methods. This can be done by using several techniques. The most common technique is the linear approximation, where the original data range is transformed to the interval of [0,1]. Thus, triangular or trapezoidal membership functions may be used in this case. In the linear regression method, one assigns the minimum and maximum attribute values. Assuming that the distribution of parameters provides a triangular membership function for the estimated parameter, the maximum value may thus be assigned as the average value of the obtained results. This may, however, cause some loss of information and bad convergence. The second technique uses bell shaped functions. The initial values of parameters can be derived from the statistics of the input data. Further, the polynomial approximation of data, either ordinary or Chebyshev, may be

used. This technique is justified by a sufficiently large number of results or by increasing the order of polynomials; however, the latter may lead to a weak generalization of results. Another approach to defining the shape of the membership function involves the use of the probability density function. The last mentioned technique was discussed in the given context more thoroughly in literature (Kostek 1999, 2003).

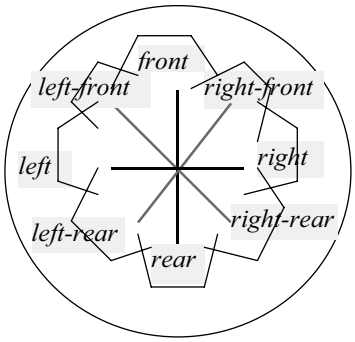


Fig. 6.6. Directivity membership functions on the horizontal plane

Intuitively, it seems appropriate to build the initial membership function by using the probability density function and by assuming that the parameter distribution is trapezoidal or triangular. The estimation of the observed relationships is given by the function shown in Fig. 6.7.

The f_1 membership function from Fig. 6.7 is defined by a set of parameters: A, a, b, c, d and is determined as follows:

$$f_1(x,A,b,c,d) = \begin{cases} 0 \quad \text{if} \quad x < a \text{ or } x > d \\ A(x-a)/(b-a) \text{ if} \quad a \le x \le b \\ A \quad \text{if} \quad b < x < c \\ -A(x-d)/(d-c) \text{ if} \quad c \le x \le d \end{cases} \tag{6.1}$$

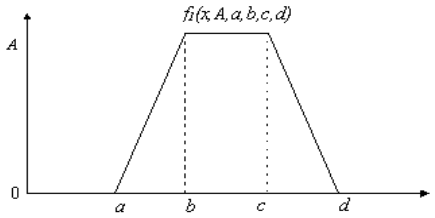


Fig. 6.7. Trapezoidal membership function estimated by the probability density function

The equation describing the *m*th moment of the probability density for the function $f_1(x,A,a,\ b,c,d)$ is calculated as follows:

$$m_n = \int_{-\infty}^{+\infty} x^n f_2(x) dx \tag{6.2}$$

The estimate of the *m*th moment of the probability density function from the test (assuming that all observation instances fall into the interval *j*, where: $j=1,2...k$) is calculated according to the formula:

$$\hat{m}_n = \sum_{j=1}^{k} x^n (P(x = x_j)) \tag{6.3}$$

where: $P(x=x_j)$ represents the probability that the attribute value of instance *x* falls into the interval *j*.

Next, the subsequent statistical moments of the order from 0 to 4 for this function should be calculated. Then, by substituting the observed values into Eq. (6.3), the consecutive values of m_n are calculated. From this, the set of 5 linear equations with 5 unknown variables *A*,*a*,*b*,*c*,*d* should be determined. After numerically solving this set of equations, the final task of the analysis will be validation of the observed results using Pearson's χ^2 test with *k*-1 degrees of freedom (Kostek 1999).

Using the above outlined statistical method, a set of fuzzy membership functions for the studied subjective sound directivity can be estimated.

Inter-Modal Testing Phase

In order to proceed with testing the inter-modal relation between sound localization and video images, another questionnaire should be used. This time, the experts' task would be assigning the crisp angle value to the incoming sound excerpt while watching a TV screen. Having previously estimated membership functions, it would be then possible to check whether the observation of video images can change sound localization and if yes then to what degree. This can be done by performing a fuzzification process. The data representing the actual listening tests would then pass trough the fuzzification operation in which degrees of membership should be assigned for each crisp input value.

The process of fuzzification is illustrated in Fig. 6.8. The pointers visible in this figure refer to the degrees of membership for the precise value of localization angle 335°. Thus, this value belongs, respectively, to the

left-rear fuzzy set with the degree of 0, to the *left-front* fuzzy set with the degree of 0.65 and to the *front* set with the degree of 0.35. The same procedure should be applied to every sound-image instance.

Consequently, the process of fuzzy inference can be started, allowing to find winning rules. The (examples) fuzzy rules have the following form:

1. if *front* AND *FRONT* than *no_shift*
2. if *left_front* AND *LEFT_FRONT* than *no_shift*
3. if *left_front* AND *FRONT* than *slight_shift*
4. if *left* AND *LEFT* than *no_shift*
5. if *left* AND *LEFT_FRONT* than *slight_*

.....

where small italic labels denote current directivity indices and the capital italic labels denote the directivity of the same sound played back during the previous tests (in the absence of vision).

It was assumed that the presence of vision is causing the shifting of sound localization to the front direction only (not to opposite directions in relation to the frontal one) and there is no possibility for phantom sources to migrate from the left to the right hemisphere and vice versa. These assumptions have been justified in practice. The rules applying to the right: front lateral and rear directions are similar to above ones. The AND function present in the rules is the '*fuzzy and*' (Kosko 1997). Thus it chooses the smaller value from among these which provide arguments of this logical function. The consequences: *no_shift*; *slight_shift*; *medium_shift*; *strong_shift* are also fuzzy notions, so if it is necessary to change them to the concrete (crisp) angle values, a defuzzification process should be performed basing on the output prototype membership functions.

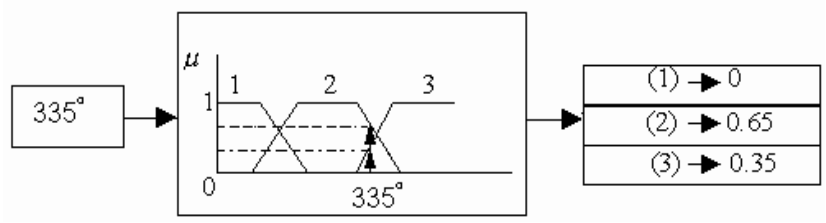


Fig. 6.8. Fuzzification process of localization angle: (1) - *left-rear*, (2) - *left-front*, and (3) - *front membership functions*

All rules are evaluated once the fuzzy inference is executed and finally the strongest rule is selected as the winning one. These are standard procedures related to fuzzy logic processing of data (Kosko 1997). The winning rule demonstrates the existence and the intensity of the phantom sound

source shifting due to the presence of vision. Since the fuzzy rules are readable and understandable for human operators of the system, thus this application provides a very robust method for studying complex phenomena related to the influence of vision coming from frontal TV screen on the subjective localization of sound sources in surround space. The mentioned defuzzification procedure (Kosko 1997) enables to map a fuzzy descriptor to a crisp angle measure every time it is necessary to estimate such a value.

Audio and video interact with each other. Mechanisms of such interaction are currently investigated in two domains: perceptual and aesthetic ones by employing fuzzy logic in the process of analysis of tested subjects' answers. The results of such experiments could yield the recommendations to sound engineers producing surround movie sound tracks, digital video and multimedia.

6.2 Genetic Algorithm-Based Processing

6.2.1 Problem Overview

The questionnaire form for assigning arrival directivity of the sound in mapping tests is shown in Fig. 6.9. In addition, 10 abstraction tests and 15 or 20 high level-abstraction tests were presented to experts' in the experiments. The abstraction tests used simple objects instead of complex ones in order to discover and to describe basic mechanisms underlying the audio-visual perception. Audio-visual presentation time schedule is shown in Fig. 6.10.

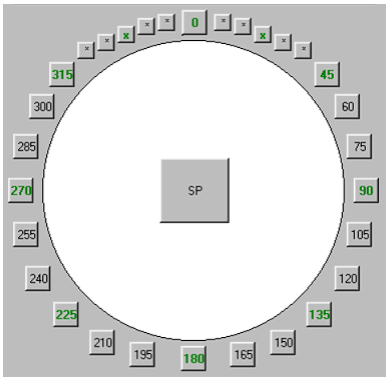


Fig. 6.9. Questionnaire form for assigning the directivity of sound arrival

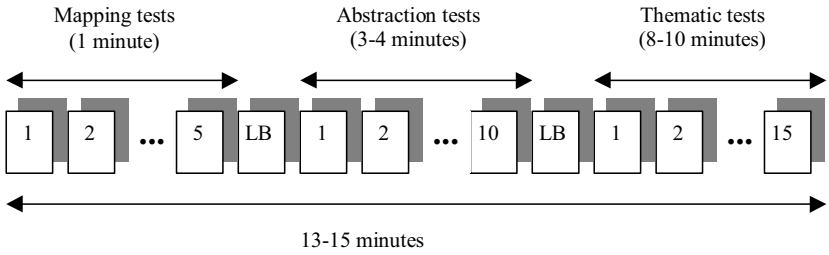


Fig. 6.10. Audio-visual material presentation time schedule, where: LB – a long break (duration of 5 to 10 seconds) made after each presentation (short pause lasted 3 seconds)

Surround sound systems allow creating phantom sound sources in 360^0 range. However using too many sound sources may introduce some errors due to inaccuracy in positioning the phantom sound sources. Thus, the number of sources was limited to the following angles: 0 (central loudspeaker), 22.5^0, 45^0 (front right loudspeaker), 90^0, 135^0 (rear right loudspeaker), 180^0, 225^0 (rear left loudspeaker), 270^0, 315^0 (front left loudspeaker), 338^0. In purpose to increase the number of possible answers experts could choose also other angles: 7.5^0, 15^0, 30^0, 37.5^0, 60^0, 75^0, 105^0, 120^0, 150^0, 165^0, 195^0, 210^0, 240^0, 255^0, 285^0, 300^0, 322.5^0, 330^0, 345^0, 352.5^0. Furthermore, in order to allow an expert to express more spatial-like impressions - not only those angle-oriented, but also some angle-group oriented entities were added, such as: L+C+R – wide central base (315^0+0^0+45^0), WF – wide front base (315^0+45^0), WR –wide right base (45^0+135^0), WB – wide back base (135^0+225^0), WL – wide left base (225^0+315^0), SS – Sweet Spot, ALL – all five channels playing simultaneously. In this way attributes defining the sound domain space were assigned.

The visual domain space was described with only one attribute assigned to thematic tests indicating whether video was present or not. In the abstraction tests several attributes were added describing for example how the line was moving on the screen: L2R – from left to right, R2L – from right to left, D2U – up, U2D – down. All those given attribute sets served as the basis for determining the structure of decision rules discovered by the data mining system.

It is important to point out that the assumption was made that all the parameters in both visual and sound domains could contain only binary data. This means that a given angle could be either completely included in the perception of a surrounding sound or completely excluded from it. Similarly, images could be used for a given test or not.

6.2.2 Knowledge Base

As a result of listening tests thirty files containing results of the abstraction tests (ten samples each) and thirty files with results of the subject tests (fifteen or twenty samples each) were obtained. One set of the subject tests answers was rejected after taking the mapping tests results into account. Finally, 300 abstraction tests and 465 thematic tests were analyzed (Czyzewski et al 2001; Smolinski and Tchorzewski 2001)

Methodology based on searching for repetitive patterns in data and generating association rules from those patterns was used for data mining in this research study. Data were represented as a simple information system. An example of a data record from the information system is shown in Fig. 6.11. A record in the abstraction test database (Fig. 6.11) contains values of 1 at 4th, 11th, 13th and 22nd positions (0 s elsewhere). This means that a sound stimulus presented at 90° (4th attribute) accompanied by an image (11th attribute) of a vertical line moving from the right to the left side of the screen was actually localized by an expert at 45° (22nd attribute).

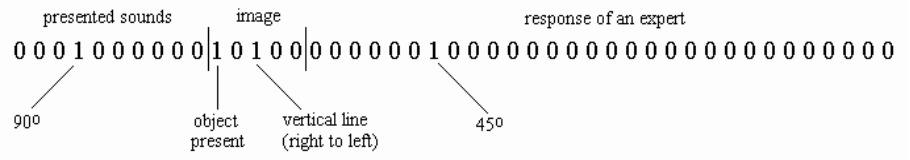


Fig. 6.11. Example of a record in the database (abstraction tests case)

After creating the appropriate data sets, it was possible to explore and analyze the data. The aim was to discover the influence of visual stimuli on the perception of a sound in surround space, thus searching for association rules was performed. The genetic algorithm was employed to perform this task. Since genetic algorithms belong to the most often used soft computing methods, thus their principles will not be reviewed here.

In this research study, the chromosomes that are being produced and modified during the evolution process represent patterns covering records in the data set. Each one of them has the length of the number of attributes describing the data (specific for the type of the tests – abstraction vs. thematic), and the alleles of a chromosome are constrained by the domains of those attributes. An allele of such chromosome can either contain a value that is valid for a corresponding attribute in the data set which is in this case 1 s (all 0 s can be omitted since such a testing is aimed at interrelation of angle and image) or a ‘don’t care’ asterisks which means that this attribute is not important and will not be used to generate a rule (Czyzewski

et al 2001; Kostek 2003). An example of a chromosome is presented in Fig. 6.12.

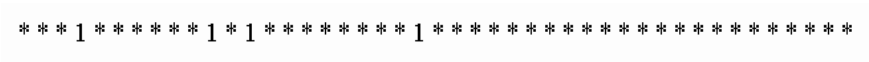


Fig. 6.12. Example of a chromosome (set positions – 4th, 11th, 13th, 22nd)

Each of such patterns has a possible coverage in data (support) which is given by the number of records matching the pattern (i.e. having given values at the 'set' position). For the example above it will be all records containing '1' at 4th, 11th, 13th and 22nd positions regardless of other values. Obviously, one should look for patterns that have relatively high support and this can form the basis of the fitness function used for this algorithm. The desired level of support in data can be adjusted by setting the *epsilon* value, which stands for the percent-based, maximum allowed error in terms of pattern coverage (the higher epsilon, the lower minimum support required) (Czyzewski et al 2001; Kostek 2003).

Although the support of a pattern is a basic feature of the fitness function implemented in the algorithm, it cannot be its ultimate characteristic. The number of 'set' positions (not the '*don't care*' asterisks) is also very important. For example, a pattern consisting only of asterisks will gain support of 100% of the data records, but it has no meaning in terms of knowledge discovery. The structure of the *IF–THEN* rules generated afterwards is also very important, and from the practical point of view patterns must contain at least two (or even three) set attribute values in order to stand as a basis for any useful association rules. Such a rule should have the following structure:

$$\{\text{presented sound}\} \cup \{\text{image}\} \Rightarrow \{\text{response of an expert}\}.$$

Obviously, not all the chromosomes will have a physical coverage in the available data set. Some of them (especially the ones with a relatively large number of set positions) might not have a support at all, however some parts of them (subsets of values) still can be very useful and after an application of some genetic operators (i.e. crossover and mutation) may produce desired results. It is crucial then to appropriately treat all those chromosomes and assign them some "credit" in terms of the fitness function even though they do not have support in data as a whole.

Based on the above discussion all the chromosomes (potential solutions) should be awarded or punished according to the specified criteria during the evolutionary process. Thus the fitness function can be completely described as a multi-layer estimation of the fitness of chromosomes in terms

of their partial support in the data at first, and then in terms of total coverage of the data weighted by the number of set positions.

Another very important feature of the genetic algorithm used here is a multi-point crossover option. In many experiments mining patterns in different types of data, this approach was found to be much more effective with regard to both the number of discovered patterns, and the time of convergence. On the basis of empirical premises the maximal number of cuts (crossover points) was set to 1 for every 10 attributes. In the example given in Fig. 6.13 there are three crossover points and the arrows point out to genetic material that will be exchanged and thus will create two new chromosomes.

Fig. 6.13. Example of a multi-point crossover (three point)

As an outcome of several evolutions modeled by this genetic algorithm, a set of data patterns was created. Those patterns along with the information about the level of their support were then used as an input to the application generating association rules.

Association rules determine the existence of some relations between attributes in data or values of those attributes. Basically they are simple *IF–THEN* type rules that, for a binary domain of values, can be considered as statements (Czyzewski et al 2001, Kostek 2003).

"*if attributes from the premise part of the rule have values of* 1 *then the attributes included in the consequent part also tend to have value of* 1".

In the discussed case, rules should be of the following type:

"*if a given set of angles was used for the reproduction of a sound and image was/was not present then the experts tended to localize the sound source at a particular angle/set of angles*".

Association rules are characterized both by their support in data (the number of cases that a given rule applies to – how "popular" the rule is) and the confidence (the ratio of the support of the rule to the number of cases that contain its premise part – revealing how sure one can be that judging on the basis of the values from the premise the rule is correct).

An algorithm of searching for association rules consists of two parts: searching for patterns hidden in data (in this project achieved initially by

the application of a genetic algorithm) and generating rules based on those patterns. The idea of the algorithm for rule generation in this research study is relatively simple. Basically it takes 'not asterisk' values of each of the patterns, divides them into subsets, and by moving those subsets from the premise to the consequent part (according to the specified constraints) creates all possible rules based on the given pattern. The algorithm is quite resource consuming, thus it removes all records that are covered by any others (i.e. those that are subsets of another set). This decreases the computational complexity of the algorithm and together with the support and confidence parameter limits the number of generated rules.

6.2.3 Pattern Searching

In order to increase the variety of patterns, the algorithm was launched on several computers simultaneously. Because of the randomness aspects of genetic algorithms, the results differed from each other. However, some of those results were duplicated.

The support threshold of desired patterns was lowered to 5%. This seems to be extremely low, but it is valid because rules based on patterns with relatively small support in data may still have quite a large level of confidence. As a result of several evolutions of the Genetic Algorithm, a total of 806 distinct patterns for the abstraction test, and 890 for the subject test were found. Some of those patterns were characterized by including set values only in the range of generated locations (angles), and not the ones that were a response from an expert. This was quite obvious, taking into consideration the fact that a great part of the generated tests consisted of different angles at the same time (e.g. WF, L+C+R, etc. – see the description of the sound space), and an appropriately engineered algorithm should definitely find them. However, some patterns that were satisfactory in terms of the rule definition were also discovered (i.e. they consisted of generated locations that were perceived by an expert, as well as of the information about the image presented). Some examples of those patterns are given below:

(Abstraction tests; *ABSTRACT* space – 45 attributes):

{1 * 1 * * * 1 * 1 * * * * * * * 1 * * * * * * * * * * 1 * * * 1 * * * * * 1 * * * * * * * *}

support: [18/300]

(Thematic tests; *THEMATIC* space – 41 attributes):

{1 1 * * * 1 1 * 1 * * * * * * * * * * * * * * * 1 * * * * * * * * * * * * * * * * * * *}

support: [16/465]

6.2.4 Rule Generation

Patterns discovered and prepared in the previous step were then used as a basis for associative rule generation. At this level, sets of attributes were divided into premises (generated angles along with the information about an image) and consequent (response from an expert) parts. After duplicated patterns have been removed, 49 effective patterns for the abstraction and 23 for the thematic tests remained. On the basis of this final set of patterns, a number of rules of a given support and confidence was generated. Some of the rules indicated a lack of any influence of the image on the perceived localization of sound, and this was usually associated with sounds perceived from behind of the listener. Nevertheless, most rules proved an existing interrelation between the auditory and visual senses. A sample of such a rule is presented below for the case of abstraction tests; this is a rule with a clear indication of audio-visual dependencies:

IF i045=1 AND i135=1 AND P=1 THEN 45=1 [s=6%] [c=66%]
(IF sound is presented at the angles of 45° and 135° and the image is present THEN the perceived angle is 45° WITH support of 6% and confidence of 66%)

IF i225=1 AND i315=1 AND P=1 AND D2U=1 THEN 315=1 [s=4%] [c=75%]
(IF sound is presented at the angles of 225° and 315° and there is an image of a horizontal line moving from down up THEN the perceived angle is 315° WITH support of 4% and confidence of 75%)

Based on the performed experiments it may be said that rules generated by the genetic algorithm proved an existence of a so-called proximity shift while perceiving sound in the presence of a video image. However the support associated with such rules is so low that it is difficult to conclude whether these rules are valid, even the confidence related to such rules is quite high. That is why in the next paragraph another approach to processing data obtained in subjective tests will be presented. It concerns a hybrid system consisted of neural network modules and rough set-based inference.

6.3 Rough-Neuro Processing

6.3.1 Neuro-Rough System Principles

As mentioned before, at least three factors should be taken into account while testing surround sound perception accompanied by video. They are such as follows: sound arrival angle, distance, and level of the sound. It is obvious that all these three might be interrelated. However, as was shown in the previous study, employing subjective tests based on fuzzy logic techniques (Kostek 1999) was sufficient to work with a single function separately and then, to interrelate these factors in some rule premises (Czyzewski et al 2000a, 2000b; Kostek 2003). Rule premises contain the above mentioned factors and assessed descriptors assigned to them during subjective testing sessions, and the consequence (decision) that resulted from these test data. However rules that were formulated were hard to verify by experts. Therefore, in this study a new concept of rule discovering was conceived. For this purpose a modular neuro-rough system was engineered that is described further on (Fig. 6.14). As mentioned in Chapter 3, such a hybrid approach is already well-adopted in applications in many fields (e.g. Chakraborty 2000; Komorowski et al 1999; Pawlak and Skowron, 1994; Peters et al 2000; Polkowski and Skowron 2000; Skowron et al 2000; Szczuka 2002).

As seen from Fig. 6.14 the main two blocks of the neuro-rough system are related to data processing. These are neural network modules, that quantize numerical data, and the rough set-based engine that extract rules from data. The elements of the input vector shown in Fig. 6.14 are numbers representing the realm of angles, distances and sound loudness values, whereas the rough set-based decision system requires quantized data. For example let us consider spherical space around the listener's head. This space can be sampled at different elevations (from below the horizontal plane to directly overhead). In addition at each elevation a full 360 degrees of azimuth can be sampled in equal sized increments. A total of some hundred of locations can be obtained in this way. However, in the experiments only horizontal plane was considered, the division of angles was as seen in Fig. 6.14. Consequently, for quantization purposes, self-organizing neural net is proposed, similarly like in other experiments (Czyzewski and Krolikowski 2001). For this purpose self-organizing map (SOM) introduced by Kohonen has been chosen (Kohonen 1990). This is one of the best known neural network clustering algorithms, which assigns data to one of specified subsets according to the clusters detected in a competitive

learning process. During this learning only the weight vector which is most similar to a given input vector is accepted for weight building. Since data can be interpreted as points in a metric space, thus each pattern is considered as an *N*-dimensional feature vector, and patterns belonging to the same cluster are expected to possess strong interval similarity according to the chosen measure of similarity. Typically, the Euclidean metric is used in SOM implementations (Kohonen 1990).

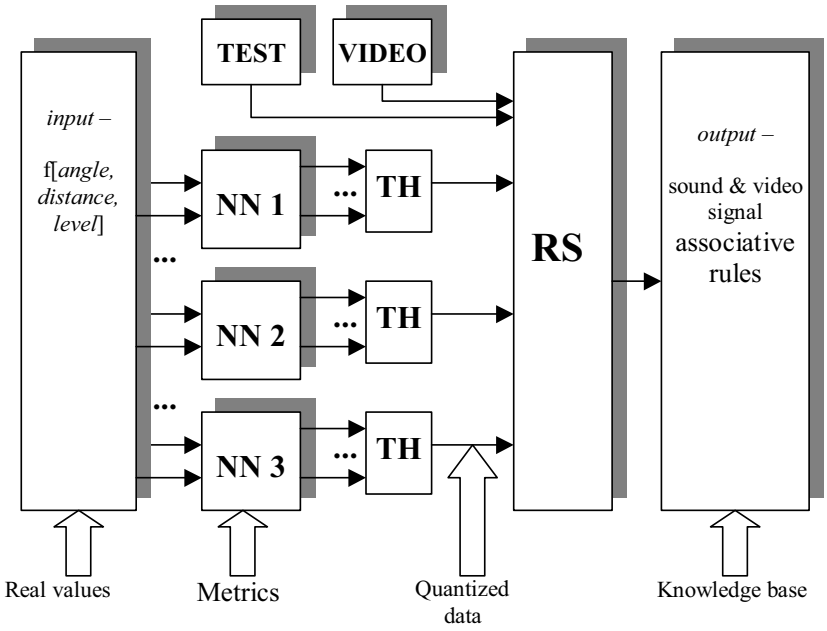


Fig. 6.14. Neuro-rough system lay-out

Using the SOM as a data quantizer, a scalar- and a vector quantization can be taken into account. In the first case, the SOM is supplied to a single element of the key vector. In the second case, a few attributes can constitute input vectors, which lowers the number of attributes helping to avoid a large number of attribute combinations in the rough set inference.

The SOM of the Kohonen type defines mapping of *N*-dimensional input data into a two-dimensional regular array of units, and the SOM operation is based on a competition between the output neurons due to any stimulation from input vector $\mathbf{x}$. As a result of the competition, the *c*th output unit wins provided the following relations are fulfilled:

$$d(\mathbf{x}, \mathbf{W}_c) = \max_{1 \le i \le K \times L} d(\mathbf{x}, \mathbf{W}_i) \text{ or } \quad c = \arg\min_{1 \le i \le K \times L} \{d(\mathbf{x}, \mathbf{W}_i)\}, \tag{6.4}$$

where $d()$ is a distance between vector $\mathbf{x}$ and weight vector $\mathbf{W}_i$ of the output neuron, whereas $K \times L$ is the dimension of the output layer.

The adaptation process can be described in terms of the minimization of an error function $E^{(k)}$, and hence the updating of the weight vectors in the kth step is performed according to the expression:

$$\bigvee_{i=1,\ldots,K\times L} \mathbf{W}_i^{(k+1)} = \mathbf{W}_i^{(k)} + \Delta\mathbf{W}_i^{(k)} = \mathbf{W}_i^{(k)} - \nabla_{\mathbf{W}} E^{(k)} \tag{6.5}$$

where initial values of weight matrix $\mathbf{W}^{(0)}$ are small random values in the range $[-1,1]$, where as the definition of error function $E^{(k)}$ is related to the concept of vector quantization, and is given by the following formula (Kohonen et al 1989):

$$E^{(k)} = \sum_{i=1}^{K\times L} h_{ci}^{(k)} \cdot \Psi\left[d(\mathbf{x}, \mathbf{W}_i)\right] \tag{6.6}$$

where h_{ci} is a spatial neighborhood kernel for the cth best matching unit. Thus, the expression updating formula can be rewritten as below:

$$\mathbf{W}_i^{(k+1)} = \mathbf{W}_i^{(k)} - h_{ci}^{(k)} \cdot \frac{\partial}{\partial \mathbf{W}} \Psi(\mathbf{x}, \mathbf{W}_c) \tag{6.7}$$

for which the adequate derivatives of function $\Psi(\mathbf{x}, \mathbf{W}_c)$ are dependent on the metrics used.

In the SOM implementation, the general form of kernel function h_{ci} is exploited, which is the Gaussian function defined as follows (Kohonen 1990):

$$h_{ci}^{(k)} = \begin{cases} \eta^{(k)} \cdot \exp\left[-\left(\dfrac{d(\mathbf{r}_c, \mathbf{r}_i)}{\sigma^{(k)}} \right)^2 \right] & ; \text{inside } N_c \\ 0 & ; \text{outside } N_c \end{cases} \tag{6.8}$$

where N_c denotes the set of neighbor nodes around the cth winner neuron, $\mathbf{r}_i$ and $\mathbf{r}_c$ are the coordinate vectors of the ith unit and the best matching one. In turn, $\eta \in [0,1]$ can be referred to as a learning rate, whereas σ corresponds to the radius of set N_c, and is limited by the size of the array of the output neurons. Both are decreasing functions of time, of which definitions are given below (Kohonen 1990).

The learning rate η is expressed by the relationship:

$$\eta^{(k)} = \alpha^{(k)} \cdot \left(1 - \frac{k}{k_{\max}}\right) \tag{6.9}$$

where the coefficient α varies according to the Kohonen's recommendations in the following way:

$$\alpha^{(k)} = \begin{cases} 0.95 & ;k \le k_1 \\ 0.9 & ;k_1 \le k \le k_2 \\ 0.01 & ;k \ge k_2 \end{cases} \tag{6.10}$$

Other recommendations are as follows: $k_1 = 0.02 \cdot k_{\max}$, $k_2 = 0.4 \cdot k_{\max}$, for which the maximum number of iterations is set to: $k_{\max} = \{10000;50000;100000\}$.

The radius σ of the neighborhood set N_c corresponds to the learning rate η according to relationship:

$$\sigma^{(k)} = \begin{cases} \lfloor \alpha \cdot k + \beta \rfloor & k \le k_1 \\ \lfloor \chi \cdot k + \delta \rfloor & k_1 \le k \le k_2 \\ 1 & k_1 \ge k_2 \end{cases} \tag{6.11}$$

where coefficients α, β, χ, δ are computed according to recommendations given by Kohonen (1990):

$$\alpha = -\frac{\sigma^{(0)} - 1}{k_1} \qquad \beta = \sigma^{(0)} \cdot \left(1 - \frac{1}{k_1}\right) - \frac{1}{k_1} \tag{6.12}$$

$$\chi = -\frac{1}{k_2 - k_1} \qquad \delta = \frac{k_2}{k_2 - k_1}$$

where the initial radius $\sigma^{(0)}$ is equal to the radius of the output array, i.e.: $\sigma^{(0)} = \max(K, L)/2$.

In the structure of the implemented SOM, the input and output nodes are fully connected, whereas the output units are arranged in the hexagonal lattice. The initial values for learning rate $\eta^{(0)}$ is equal to 0.95. For the purposes of the neuro-rough hybridization, at the end of the weight adaptation

process the output units should be labeled with some symbols. It is done in order to assign quantized input data to symbols which are to be processed in the rough set inference.

The engineered rule induction algorithm is based on the well described in literature rough set methodology (Komorowski et al 1999; Pawlak 1982; Pawlak 1991, Pawlak and Skworon 1994; Skowron et al 2000). The used algorithm aimed at reducing the computational complexity (Czyzewski and Krolikowski 2001). This pertained reducing the values of attributes and searching for reducts, so that all combinations of the conditional attributes are analyzed at reasonable computational cost. Particularly, for a given sorted table, the optimum number of sets of attributes A ($A \subseteq C = \{a_1, \ \ldots, \ a_i, \ \ldots, \ a_{|C|}\}$), subsets of conditional attributes C, can be analyzed using special way of attribute sorting (Czyzewski and Krolikowski 2001). The algorithm splits the decision table into two tables: consisting of only certain rules and of only uncertain ones. There is an additional information associated with every object in them. The information concerns the minimal set of indispensable attributes and the rough measure μ_{RS}. The latter case is applied only for uncertain rules. Other details corresponding to the rough set-based engine can be found in literature (Czyzewski and Krolikowski 2001).

6.3.2 Experiments

Results from test sessions gathered in a database are then further processed. Table 6.3 that consists of database records can be considered as a decision table that resulted from both the abstraction and thematic tests. The type of the test is therefore one of the attributes contained in the decision table. Other attributes included in the decision table are: '*angle*', '*distance*', '*level*', '*video*' and a decision attribute called '*proximity_effect*'. To differentiate between attributes resulting from experts' answers, actual values of angles, distance and level known to the experimenter two adjectives, namely, 'subjective' and 'objective' were added, thus making it six attributes altogether. As was mentioned before, during the test session experts are asked to fill in questionnaire forms, an example of which was shown in Fig. 6.3. It should be remembered that questionnaire values are numerical, thus values indicated by experts form a feature vector that is fed through the neural networks modules. The neural network module assigns a numerical value indicated by an expert to one of the clusters corresponding to semantic descriptors of a particular attribute and returns it at the NN output. This is possible by adding a threshold function operating in the

range of (-1,1) to the system shown in Fig. 6.14. Such comparators are shown schematically behind the outputs of the NN, however in reality they are assigned to the neuron in the hidden layer. Their role is to choose the strongest value obtained in the clustering process. Therefore Table 6.3 contains descriptors obtained by a neural network-based quantization related to '*angle_subjective*', '*distance_subjective*', '*level_subjective*' attributes. Semantic descriptors related to the '*angle_subjective*' attribute are as follows: '*none*', '*front*', '*left_fron*t', '*left*', '*left_rear*', '*rear*', *right_rear*', *right*', '*right_front*'. All but one such a descriptor is obvious. The '*none*' descriptor is related to the case when distance equals 0, thus a phantom sound is positioned in a '*sweet-spot*' (expert's seat). This means that sound is subjectively perceived as directly transmitted to the head of an expert. In addition the '*distance_subjective*' is quantized by the NN module as: '*none*' ('sweet-spot'), '*close*' (large distance from the screen), '*medium*', '*far*' (small distance from the screen), and correspondingly '*level_subjective*' is denoted as: '*low*', '*medium*', '*high*'.

Values of angles, distance and level of a phantom sound source are given numerically as set by the experimenter, however they are quantified values (angles in degrees, distance in centimeters and level in dB). The range of angle attribute was already shown. The quantization resolution of angles and distance is directly related to the resolution of phantom sources created by the *Samplitude* 2496 software. In a way this limits the number of created phantom sources. Level values were quantified in the range of (50 dB to 100 dB) with a 10 dB step. The problem of the quantization of level, distance and level attributes is further complicated because of some acoustical principles, which will be however not reviewed here. These values were left as numbers because it is easier to understand the rule in such a way. On the other hand descriptors related to '*test*', '*video*' attributes and the proximity effect attributes were set as semantic descriptors. Therefore '*sound*' and '*video*' attributes can have values such as follows: '*abstraction*', '*thematic*' and correspondingly: '*no_video*', '*static_image*', '*dynamic_image*'. The decision attribute can be read as '*no_shift*', '*slight_shift*', '*medium_shift*' and '*strong_shift*' and these descriptors will appear in the consequence part of a rule.

Table 6.3. Decision table

Experts' answers	Angle_ subj.	Distance	Level	Angle_ objective	...	Test	Video	Decision - proximity_ effect
e_1	front	close	medium	0^0	...	abstrac-tion	static_ image	no_ shift
e_2	left_front	close	high	60^0	...			medium_ shift
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
e_n	left_rear	far	medium	315^0	...	thematic	dynamic _image	strong_ shift

Sample rules that can be derived from the decision table are presented below:

if *left_front* AND *close* AND *medium* AND 0^0 AND *20* AND *70* AND *abstraction* AND *static* than *no_shift*
if *left_front* AND *close* AND *medium* AND 60^0 AND *20* AND *70* AND *thematic* AND *static* than *slight_shift*
if *none* AND *none* AND *high* AND 0^0 AND *0* AND *90* AND *abstraction* AND *static* than *no_shift*
.....

where the numerical values denote the actual directivity, distance or level of the transmitted sound and related italic labels denote indices as indicated by experts and then quantized by the NN module. Other values are as explained before. Rules that will have a high value of the rough set measure can be considered as a knowledge base of associative rules for video, multimedia and film creators.

The subjective listening tests proved that visual objects could influence the subjective localization of sound sources. Measurement data showed that visual objects may "attract" the listeners' attention, thus in some cases sound sources may seem to be localized closer to the screen. It was found that the image proximity effect is listener-dependent, what is probably related to some individual psychological processes occurring in human brains.

As seen from the presented concepts and experiments, subjective descriptors and numerical values gathered in the decision table can be then processed by the rough set-based method. In this way a new concept of computing with words was presented that allow processing data obtained from subjective tests and objectively given measures.

On the basis of the experiments described in this Chapter, it can be stated that subjective tests seem appropriate for the analysis of the correlation between hearing and sight senses due to the perception of a surround sound. It creates an environment for automatic exploration of data derived from psychoacoustic experiments, and knowledge discovery based on modern, soft computing–oriented methodologies. The results of such experiments could yield the recommendations to sound engineers producing surround movie sound tracks, digital video and multimedia.

References

Bech S, Hansen V, Woszczyk W (1995) Interactions Between Audio-Visual Factors in a Home Theater System: Experimental Results. In: Proc 99th Audio Eng Soc Conv, New York, Preprint No 4096

Beerends JG, de Caluwe FE (1999) The Influence of Video Quality, on Perceived Audio Quality and Vice Versa. J Audio Eng Soc 47:355-362

Brook M, Danilenko L, Strasser W (1984) Wie bewertet der Zuschauer das stereofone Fernsehes? In: Proc 13 Tonemeistertagung Internationaler Kongres, pp 367-377

De Bruijn W, Boone (2002) M Subjective Experiments on the Effects of Combining Spatialized Audio and 2D Video Projection in Audio-Visual Systems. In Proc 112 Audio Eng Society Conv, Munich, Germany

Chakraborty B (2000) Feature Subset Selection by Neuro-Rough Hybridization. In: Proc 2nd Int Conf on Rough Sets and Current Trends in Computing (RSCTC'2000), Banff, pp 481-487

Czyzewski A, Kostek B, Odya P, Zielinski S (2000a) Influence of Visual Cues on the Perception of Surround Sound. In: Proc 139th Meeting of the Acoustical Society of America, J Acoust Soc Am 107: 2851, Atlanta

Czyzewski A, Kostek B, Odya P, Zielinski S (2000b) Determining Influence of Visual Cues on the Perception of Surround Sound Using Soft Computing. In: Proc 2nd Int Conf on Rough Sets and Current Trends in Computing (RSCTC' 2000), Banff, pp 507-514

Czyzewski A, Kostek B, Odya P, Smolinski T (2001) Discovering the Influence of Visual Stimuli on The Perception of Surround Sound Using Genetic Algorithms. In: Proc 19th Int Audio Eng Soc Conference, Germany, pp 287-294

Czyzewski A, Krolikowski R (2001) Neuro-Rough Control of Masking Thresholds for Audio Signal Enhancement. J Neurocomputing 36: 5-27

Dvorko N, Ershov K (1998) Some Aspects of Audio-Visual Perception of Different Program Materials (Film, Video, and Multimedia). In: Proc 105 Audio Eng Soc Conv, San Francisco, USA

Florek A, Szczuko P (2002) Testing of audio-visual correlation for surround sound systems and digital TV (*in Polish*). MSc thesis, Gdansk University of Technology, Kostek B (supervisor)

Gardner MB (1968) Proximity Image Effect in Sound Localization. J Acoust Soc Am 43: 163

Hollier MP, Voelcker R (1997) Objective Performance Assessment: Video Quality as an Influence on Audio Perception. In: Proc 103rd Eng Soc Conv. New York, Preprint No. 4590

Kaminski J, Malasiewicz M (2001) Investigation of influence of visual cues on perceived sound in the surround system (*in Polish*). M. Sc. thesis, Sound and Vision Eng Dept, Technical Univ of Gdansk, Kostek B (supervisor)

Kohonen T (1990) The Self-Organizing Map. In: Proc IEEE 78, pp 1464-1477

Kohonen T, Oja E, Simula O, Visa A, Kangas J (1996) Engineering Applications of the Self-Organizing Map. In: Proc IEEE 84, pp 1358-1384

Komiyama S (1989) Subjective Evaluation of Angular Displacement Between Picture and Sound Directions for HDTV Sound Systems. J Audio Eng Soc 37: 210

Komorowski J, Pawlak Z, Polkowski L, Skowron A (1999) Rough Sets: A Tutorial. In: Pal SK, Skowron A (eds) Rough Fuzzy Hybridization. A New Trend in Decision-Making. Springer Verlag, Singapore, pp 3-98

Kosko B (1997) Fuzzy Engineering. Prentice-Hall Intern Ed, New Yersey

Kostek B (1999) Soft Computing in Acoustics, Applications of Neural Networks, Fuzzy Logic and Rough Sets to Musical Acoustics. Studies in Fuzziness and Soft Computing. Physica Verlag, Heilderberg New York

Kostek B (2003) Rough-Neuro Approach to Testing Influence of Visual Cues on Surround Sound Perception; Chapter 22, Rough-Neuro Computing: Techniques for Computing with Words, Pal SK, Polkowski L, Skowron A, (eds), 555-572, Springer Verlag, Series on Cognitive Technologies, Berlin, Heidelberg, New York

Pawlak Z (1982) Rough Sets. J Computer and Information Science 11: 341-356

Pawlak Z (1991) Rough Sets - Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht

Pawlak Z (2000) Rough Sets and Decision Algorithms. In: Proc 2nd Int Conf on Rough Sets and Current Trends in Computing (RSCTC'2000), Banff, pp 1-16

Pawlak Z, Skowron A (1994) Rough Membership Functions. In: Yager R, Fedrizzi M, Kacprzyk J (eds) Advances in the Dempster-Shafer Theory of Evidence. John Wiley & Sons, NewYork, pp 251-271

Peters JF, Skowron A, Han L, Ramanna S (2000) Towards Rough Neural Computing Based on Rough Membership Functions: Theory and Application. In: Proc 2nd Int Conf on Rough Sets and Current Trends in Computing (RSCTC' 2000), Banff, pp 572-579

Polkowski L, Skowron A (2000) Rough-Neuro Computing. In: Proc 2nd Int Conf on Rough Sets and Current Trends in Computing (RSCTC' 2000), Banff, pp 25-32

Rumsey F, Ward P, Zielinski S (2004) Can Playing a Computer Game Affect Perception of Audio-Visual Synchrony? 117 Audio Eng Soc Conv San Francisco, USA

Sakamoto N, Gotoh T, Kogure T, Shimbo M (1981) Controlling Sound-Image Localization in Stereophonic Reproduction. J Audio Eng Soc 29: 794-798

Sakamoto N, Gotoh T, Kogure T, Shimbo M (1982) Controlling Sound-Image Localization in Stereophonic Reproduction: Part II. J Audio Eng Soc 30: 719-721

Skowron A, Stepaniuk J, Peters JF (2000) Approximation of Information Granule Sets. In: Proc 2nd Int Conf on Rough Sets and Current Trends in Computing (RSCTC' 2000). Banff, pp 33-40

Smolinski T, Tchorzewski T (2001) A System of Investigation of Visual and Auditory Sensory Correlation in Image Perception in the Presence of Surround Sound (*in Polish*). MSc thesis, Polish - Japanese Institute of Information Technology

Szczuka MS (1998) Rough Sets and Artificial Neural Networks. In: Pal SK, Skowron A, (eds) Rough Sets in Knowledge Discovery: Applications, Case Studies and Software Systems. Physica Verlag, Heidelberg New York, pp 449-470

Thomas GJ (1941) Experimental Study of the Influence of Vision of Sound Localization. J Exp Psych 28: 163-177

Witkin HA, Wapner S, Leventhal T (1952) Sound Localization with Conflicting Visual and Auditory Cues. J Exp Psych 43: 58-67 (citation after Sakamoto)

Wladyka M (1987) Examination of Subjective Localization of Two Sound Sources in Stereo Television Picture (*in Polish*). MSc thesis, Sound Eng Dept, Technical Univ of Gdansk

Woszczyk W, Bech S, Hansen V (1995) Interactions Between Audio-Visual Factors in a Home Theater System: Definition of Subjective Attributes. In: Proc 99th Audio Eng Soc Conv, New York, Preprint No 4133

Zielinski S, Rumsey F, Bech S (2003) Comparison of Quality Degradation Effects Caused by Limitation of Bandwidth and by Down-mix Algorithms in Consumer Multichannel Audio Delivery Systems. In: Proc 114th Audio Eng Soc Conv, Amsterdam, Netherlands

7 CONCLUDING REMARKS

The choice of problems presented in this study is intended to emphasize that in some cases even the classical problems of acoustics can be addressed and solved by means of new methods, especially those arising from the soft computing domain. Before soft computing methods were introduced, all applications dealing with uncertainty were based on the probabilistic approach. Meanwhile, in the case of some of the studied applications, such as automatic recognition of musical phrases, it is impossible to base the research on such an approach only, because each musical phrase has its unique character that cannot be sufficiently described by any statistics. Similarly, the statistical processing of subjective testing results is not fully reliable in most practical applications in which relatively small data sets are available. Moreover, the hitherto used statistical analyses do not allow for directly formulating rules showing the relations between assessed parameters. Such rules are needed to analyze the acoustical phenomena underlying the preference of subjective quality of sound. In the above mentioned applications a rule-based decision systems are necessary to ensure a more accurate data analysis and a better understanding of the phenomena under scrutiny on the basis of obtained results.

Rough set-based systems are generally known for that they can generate rules from data sets and, what is of paramount importance, because they provide ways for handling data with internal inconsistencies. These features proved to be of significant importance to presented applications, as subjective assessment of musical patterns made by experts is usually highly inconsistent. Moreover, the traditional statistical analysis of subjective test results cannot reveal hidden relations between tested parameters nor can it provide the instructions on how to tune a system based on such parameters. Consequently, the rough set method was extensively used which is one of the most advanced and well-developed data analysis techniques available today, offering effective tools to extract knowledge from data. In some applications also the fuzzy logic proved to be applicable to such problems as subjective quantization of parameter ranges or calculating global subjective preference on the basis of such operators as fuzzy union and fuzzy intersection. The fuzzy logic also helped to solve the prob-

Bożena Kostek: *Perception-Based Data Processing in Acoustics*, Studies in Computational Intelligence (SCI) **3**, 419–420 (2005)
www.springerlink.com

lem of musical instrument control – so far impossible to overcome by means of crisp logic.

The experiments conducted within the framework of this research work encompassed the implementation of selected computational intelligence methods for the purposes of acquiring and recognizing musical signals and phrases. These methods were also applied to verify subjective acoustical assessments. The problems posed were solved through the use of neural networks, fuzzy logic and rough set-based methods, genetic algorithms, and hybrid decision-systems.

The research results obtained during the course of the work confirm the viability of algorithms from the computational intelligence area for solving problems of musical acoustics and also psychophysiology of hearing. These problems, due to their complexity as well as to the unrepeatable nature of acoustical phenomena, escape analyses that are based on deterministic models. Some sample results of the musical instrument class recognition and musical duet separation were shown. Such experiments belong to the so-called Musical Information Retrieval field, which aim at automatic retrieval of complex information from musical databases. It was shown that the soft computing approach to music instrument classification is justified with recognition scores. Usually scores obtained for a small number of instruments are very high, and for a larger number of instruments, in most cases, despite the decision vagueness, the system indicates the appropriate instrument. In addition, as seen from the results, the Frequency Envelope Distribution (FED) algorithm separates musical duets quite efficiently. It is worth noting that after the sounds have been processed and separated, human experts recognized them without any difficulties. Further experiments will include some optimization of the FED algorithm in order to improve the ANN-based recognition process results.

Other case studies presented in this book encompassed applications of soft computing methods to the area of psychophysiology with a dedication to hearing problem solutions. Looking at the results obtained, it may be concluded that such an approach proved to be fully justified. The last presentation concerned audio-visual correlation. This problem should be further explored since multimodal approach may help to improve performance of many algorithms and at the same time may give a new insight into integration and interaction of human perception.