ARTIFICIAL INTELLIGENCE-BASED APPROACH TO MODELLING OF PIPE ORGANS

B. Hamadicharef

Ph.D. December 2005

ARTIFICIAL INTELLIGENCE-BASED APPROACH TO MODELLING OF PIPE ORGANS

by

BRAHIM HAMADICHAREF

A thesis submitted to the University of Plymouth
in partial fulfillment for the degree of

DOCTOR OF PHILOSOPHY

School of Computing, Communications and Electronics
Faculty of Technology

In collaboration with
Musicom Ltd

December 2005

Artificial Intelligence-based Approach to Modelling of Pipe Organs

Brahim Hamadicharef

Abstract

The aim of the project was to develop a new Artificial Intelligence-based method to aid modeling of musical instruments and sound design. Despite significant advances in music technology, sound design and synthesis of complex musical instruments is still time consuming, error prone and requires expert understanding of the instrument attributes and significant expertise to produce high quality synthesised sounds to meet the needs of musicians and musical instrument builders. Artificial Intelligence (AI) offers an effective means of capturing this expertise and for handling the imprecision and uncertainty inherent in audio knowledge and data.

This thesis presents new techniques to capture and exploit audio expertise, following extended knowledge elicitation with two renowned music technologist/audio experts, developed and embodied into an intelligent audio system. The AI combined with perceptual auditory modeling based techniques (ITU-R BS 1387) make a generic modeling framework providing a robust methodology for sound synthesis parameters optimisation with objective prediction of sound synthesis quality. The evaluation, carried out using typical pipe organ sounds, has shown that the intelligent audio system can automatically design sounds judged by the experts to be of very good quality, while significantly reducing the expert's work-load by up to a factor of three and need for extensive subjective tests.

This research work, the first initiative to capture explicitly knowledge from audio experts for sound design, represents an important contribution for future design of electronic musical instruments based on perceptual sound quality, will help to develop a new sound quality index for benchmarking sound synthesis techniques and serve as a research framework for modeling of a wide range of musical instruments.

Author's Declaration

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other University award.

Publications:

- Hamadicharef, B. and Ifeachor, E. C. "Artificial Intelligence based Modeling of Musical Instruments", in Proceedings of the 2nd COST-G6 Workshop on Digital Audio Effects December 9-11, 1999, Trondheim, Norway.
- Hamadicharef, B. and Ifeachor, E. C. "An Intelligent System Approach to Sound Synthesis Parameter Optimisation", in Proceedings of the 111^{th} Audio Engineering Society convention, November 30 - December 3, 2001, New York, USA. Preprint 5484.
- Hamadicharef, B. and Ifeachor, E. C., "Objective Prediction of Sound Synthesis Quality" in Proceedings of the 115^{th} Audio Engineering Society convention, October 10-13, 2003 New York, USA. Preprint 5958.

Signed .....

Date 12/04/06

Acknowledgements

I want to express my profound gratitude to my supervisor and director of studies, Professor Emmanuel C. Ifeachor for his supervision, patience, guidance and encouragement during all the phases of my research, and for the benefit of his wide engineering rigour and research vision.

I would like to thank Dr Robin Clark (Allen & Heath Ltd) for introducing me to Tony Koorlander and Graham Blyth of Musicom Ltd (Bideford, Devon) who have shared their vast knowledge and support. I would like to thank my research colleagues and old friends: Dr Lingfen Sun, Mr Zhizi Qiao, Dr Julian Tilbury, Dr Levente Tóth, Dr Nicolas Outram, as well as new ones Nicolai Heilemann and Michael Hess.

Finally, my warmest thanks to my parents and family to whom I am particularly indebted for their monumental, unwavering support and encouragement on all fronts. They have truly always been there for me, and without them none of this would have been possible.

Brahim Hamadicharef

Dedication

to my family and friends

Brahim Hamadicharef

December 19, 2005

Table of Contents

List of Tables

List of Figures

Glossary

AHC	Agglomerative Hierarchical Clustering
ASIC	Application-Specific Integrated Circuit
DI	Distortion Index
DUT	Device Under Test
DSP	Digital Signal Processing
FFT	Fast Fourier Transform
FM	Frequency Modulation
GAS	Group Additive Synthesis
MOVs	Model Output Variables
MSE	Mean Square Error
NMR	Noise-to-Mask Ratio
ODG	Objective Difference Grade
PEAQ	Perceptual Evaluation of Audio Quality
PESQ	Perceptual Evaluation of Speech Quality
PSNR	Power of Signal-to-Noise Ratio
PVoc	Phase Vocoder
RMS	Root Mean Squared
SDG	Subjective Difference Grade
STFT	Short-Time Fourier Transform
TEO	Teager Energy Operator

Chapter 1

Introduction

1.1 Background

Digital synthesis of musical instrument sounds is a key aspect of the rapidly growing field of computer music technology. Sound synthesis makes it possible to create music and sound beyond the physical boundaries of acoustic musical instruments. This is important as it informs the design and use of musical instruments and aids the preservation of historical instruments. In principle, sound synthesis has significant advantages over other methods of digitally re-creating musical instruments as it can preserve the fine detail that is paramount in producing musically acceptable renditions. In particular, it is an efficient way of producing realistic sound by storing relevant sound information in real time accessible form, enabling continuous 'animation' in an acoustic environment, just like the real instrument.

However, despite significant advances in music technology and digital signal processing, sound design and synthesis for complex musical instruments, such as pipe organs for example, require an understanding of the attributes of the instrument (e.g. how the resonance of individual pipes contribute to the dynamic pitch variation and character of the overall sound and what sound features to use during synthesis to retain these) and significant expertise to produce synthesised sounds of sufficient quality to meet the needs and desires of musicians/musical instrument builders. Some of the best results are still

found empirically and the process is time-consuming because of the enormous complexity and size of the data structure. Usually, the option taken is to reduce the data size in order to make it manageable, but this results in a poor re-synthesis model, and disappointing final instrument modeling.

The problem exists in two stages. Firstly, data collection/analysis requires skills and understanding only gained over the years. Second, the data needs to be auditioned and manipulated in order to refine the results in the re-synthesised instrument. These data must be represented in a form that allows 'transportability' between 'instruments' and easy manipulation via parameters that an instrument builder or musician would understand and relate to. Semi-automatic, analysis software tools exist to assist sound design and synthesis, but these always require skilled user intervention in order to achieve acceptable results. The process is extremely time consuming and error prone, restricting productivity.

1.2 Audio Problems

The major problems in computer music are the following:

- **Difficulty in producing realistic sound.** Over the last 25 years, many sound synthesis techniques have been invented, refined and improved [19][148]. All aim to produce musical sounds as close as possible to the original acoustic version. Most techniques lack realism in the sound they produce and experienced musicians can usually quickly determine if the sound is real or synthetic. Over the last decade, a lot of interest has been shown for modeling musical instruments based on their acoustic features. Physical modeling is a recent technique that emulates the acoustic of the instrument using the theory of propagation of waves, for example pressure waves in the tube of a flute or the vibrations of the strings in the case of a guitar.

- **Difficulty in producing realistic articulation.** Producing high quality sounds

that best resemble the acoustic version is possible; making the articulation between the sounds is more difficult. Most recent modeling techniques focus on the instrument itself, from an acoustics point of view, rather than trying to find a technique that generates the sound, solve this problem of realistic sound articulation as the model behaves like the instrument. This new approach has shown success in both academic research community and in industry (e.g. Yamaha VL1-m). Still a lot of research efforts are needed in this direction.

- **Model parameters optimisation**. Finding the optimal parameters of synthesis models to produce the desired sound can be a real challenge. Some synthesis models are simple (additive) whereas others can be much more complex physical models. The design process of high quality sound is tedious and very time consuming, involving a long iterative "trial and error" procedure. It can be viewed as a "search problem" in which the goal is to find an optimum point in a large solution space. Many attempts to tackle this problem using artificial intelligence techniques have been found in the literature such as using Genetic Algorithms to design Hybrid sampling plus wavetable synthesis [174], using ideas from natural selection to evolve synthesized sounds [166] and using a genetic annealing algorithm for Double Frequency Modulation (DFM) synthesis [112].

- **Lack of audio knowledge**. Until now, most of the modeling of musical instruments has been largely on the synthesis of specific sounds and not on the instrument itself. However, musical instruments can produce a wide variety of sounds and only a deep understanding of the acoustics of that instrument can help improve modeling. There is a real need to collaborate with audio professionals such as musical instrument manufacturers (people who make acoustic musical instruments), musicians (people who play instruments) and audio engineers (people who record the sound), to capture audio expertise. Moreover, expertise in sound synthesis is also

required to fully understand the sound model, its advantages and limitations, and to make the best of the current capabilities of modern technologies.

- **Audio quality**. At present, modeling of musical instruments is based on the analysis of sound recordings from acoustic musical instruments, and possibly the one considered to be the best like a *Stradivarius* violin or *Steinway* piano. The sound quality of these recordings is very important as they are used as reference, and for the re-synthesised sound to be indistinguishable from the original one. Recently, techniques have been developed trying to consider perceptual issues and to take into account the way humans perceive sounds and music. Developments of audio coding techniques such as MP3 (MPEG Layer 3) [17][16] have been real motivators with application such as audio streaming over the Internet. However, there is a need to assess, objectively, the quality of synthetic sounds and to develop an objective quality index for sound synthesis.

- **Lack of control**. Regardless of the sound synthesis technique, its control is the most important for the musician. A typical example is Frequency Modulation (FM) synthesis, which can produce a wide variety of sounds but musicians have always found it very difficult to properly use to express their musical creativity. New developments of user-friendly interfaces are essential for future computer-based music systems to be of real benefit to musicians. A high-level access to the synthesis model is required. A Graphical User Interface (GUI) must be developed, in consultation with musicians and end-users, such that their operation is kept intuitive, leaving more time for creative musical tasks. Keyboards are still the most widely used device with few alternatives, such as wind controllers [154], electric violins [69] and electric cello [124], but still limited in their real ability to control, in an effective manner, current sound synthesis models. Complex and accurate sound synthesis models of trumpets [103], for example, have been developed but still lack proper

controllers. An original design was found in [121], a *penny whistle controller* with optical sensors connected to a desktop computer, which is used to control simple digital waveguide models of wind instruments. There is still a need to develop advanced musical controllers that will allow other musicians, from the classic orchestra to use electronic versions, seeking to provide the same *feeling* of playability and potential in musical creation.

From above, it can be seen that it is of particular interest to develop models of audio expertise to provide solutions to the problems. If the expertise of instrument builders (also called luthiers) can also be captured to understand the instrument, the computer modeling techniques could be further improved. Musicians develop "golden ears" and this knowledge would allow engineers to devise systems that could assess audio quality of sounds. Furthermore, with the aid of professional musicians one would understand the limitations of current musical instruments and hence exploit this in the research, design and development of future electronic musical instruments. Modeling of audio knowledge would represent a signification contribution to computer music. From the above mentioned, it is clear that there is a real need in audio to take into account the vast expertise of experts in the field to provide efficient solution to the problems. If for example the expertise of instrument builders could be captured and exploited in a computer model, in order to understand the instrument, the modeling of musical instruments could be greater improved.

1.3 Proposed Solution

Artificial Intelligence (AI) offers the potential for encoding audio expertise in a computer model, leading to an exploitation of human understanding of musical instruments and audio quality. The challenge is to establish whether audio expertise, required for high quality modeling musical instruments, can be accurately formalized and to develop an

intelligent system in which the knowledge of audio experts would be captured and embodied in a computer taking into account the uncertainty inherent in audio engineering. A Fuzzy Expert System that exploits knowledge of audio experts should improve the current state of the art in sound synthesis and modeling of musical instruments, and provide efficient solutions to current problems. At the time of writing, this research is the first such investigation in computer music.

1.4 Aims and Objectives

The aims of this research project are to develop new techniques based on artificial intelligence to overcome the above problems, and to use these techniques to develop an intelligent system that supports modeling of acoustic musical instruments and sound design, and gives access to audio expertise. Such a system would provide, for the first time, a unique platform for demonstrating the full potential of modeling audio knowledge and for audio research.

Specific objectives are to:

- Establish a real-time audio platform for digital signal processing and sound synthesis
- Develop and evaluate real-time sound synthesis techniques
- Obtain a variety of sound data, including high quality acoustic sounds considered as reference (sound templates)
- Develop new techniques to enhance sound and assess its quality
- Determine robust and quantifiable features from the sounds and develop new techniques to extract them from the sound to allow detailed analysis
- Develop an intelligent system to automatically (or semi-automatically) classify the sound and improve the process of sound design using the analysis results

- Undertake a preliminary evaluation of the system and investigate its use in sound design with the aid of audio experts.

1.5 Overview of the thesis

Chapter 2 presents the key issues in sound synthesis and design. A brief introduction on acoustic musical instruments is given. Sound analysis and sound synthesis techniques are described. Subjective and objective methods for assessing perceived audio quality are presented, including conventional listening tests and the recent Perceptual Evaluation of Audio Quality (PEAQ) algorithm.

Chapter 3 presents an overview of the novel approach to sound synthesis parameter optimisation based on an intelligent audio system which consists of: a sound analysis engine, an audio feature processing engine, a sound synthesis engine, and a sound quality assessment engine. In Section 3.6, the optimisation loop, which allows to automate the system based on feedback from the objective measure of the audio quality, is described. The implementation of the intelligent system, a combination of C++ and MATLAB scripts integrated into Musicom Ltd analysis/synthesis tool WinPVoc, is described in Section 3.7.

In Chapter 4 details of the sound analysis engine are presented. It is based on Phase Vocoder analysis which extracts key audio features from the time-varying evolution of the harmonics, identified after knowledge elicitation sessions with the audio experts. These audio features, used to model musical instruments, are described in Section 4.3. They include some attack, decay and release timing information, both amplitude and frequency envelopes and modulations of each harmonic, the harmonic distribution, some modified Tristimulus parameters, the pitch, the brightness and a noise factor. The automatic extraction of these audio features has been developed using MATLAB scripts and is detailed in Section 4.4.

Chapter 5 first presents the development cycle of a fuzzy expert system with a brief introduction to essential elements of fuzzy logic to help understand the development of the fuzzy model of audio expertise, which is the heart of the audio features processing engine. In Section 5.5, the knowledge elicitation process is then described in detail. Its purpose is to capture rules on how the audio experts design and synthesise sounds. The design of the fuzzy sets and variables of the fuzzy inputs is presented, followed by the fuzzy rules which define how the audio experts generate sound synthesis parameters (fuzzy outputs) from manipulation of the audio features (fuzzy inputs). The implementation of the fuzzy model is described in Section 5.10, first within the MATLAB environment and secondly, with the C++/MFC† fuzzy library and fuzzy expert system developed using Microsoft Visual C++ 6.0 and linked to WinPVoc. The concept and preliminary results were first published in [60] and presented at the 111^{th} Audio Engineering Society convention in New York, in December 2001.

Chapter 6 presents the evaluation, modeling results of the novel approach to sound synthesis parameter optimisation, and performance of the intelligent audio system. In Section 6.2 the sound database and the automated sound analysis are described in detail. The methods used to assess the sound quality are presented in Section 6.3. Seven sounds, selected from the database, have been used to evaluate the intelligent system. Results of their modeling are presented from Section 6.4.2 to Section 6.4.8. Preliminary results of the fuzzy model performance are presented in Section 6.5. Section 6.6 discusses the significance of the results and benefits to the audio experts. This work has been published in [61] in the proceedings of the 115^{th} Audio Engineering Society convention (New York, October 2003).

Finally, Chapter 7 gives a summary of the research carried out, discusses the main contributions of this work, identifies the limitations of the current research and proposes future research directions. These will include extensions to the intelligent system and

†Microsoft Foundation Classes

fuzzy model, developments of an advanced sound database and novel sound quality index, and finally using the novel approach for other musical instruments and in other fields of audio research. This chapter also concludes this thesis.

Chapter 2

Key Issues in Sound Synthesis and Design

2.1 Introduction

This chapter presents the key issues in sound synthesis and design. A brief introduction to acoustic musical instruments is first given. Sound analysis techniques and sound synthesis techniques are presented. Finally, methods used for the assessment of audio quality are described in details.

2.2 Acoustic Musical Instruments

Fletcher and Rossing in [39], divide acoustic musical instruments into 3 main families: the string instruments, the wind instruments and the percussion instruments. Within each family, sub-classes can further distinguish each instrument, depending on their acoustics principles.

- **String instruments.** String instruments include sub-classes such as Guitars and Lutes, bowed string instruments, Harps, Harpsichords and Clavichords and finally the Piano.

    - **Guitars and Lutes** include guitars with acoustics related to guitar body

resonances, frets and compensations, Lutes and other plucked strings instruments.

- **Bowed String instruments** include the Violin, Viola, Cello, Double Bass as well as Viols.
- **Harps, Harpsichords and Clavichords** include the Koto, the Harp, the Harpsichord and Clavichord.
- **Piano** Piano is a sub-class on its own, with very complex acoustics concerned with the piano action, piano strings, string excitation by the hammer, the soundboard, complex interaction between the strings, bridge and soundboard, issues of tuning and inharmonicity.

- **Wind instruments.** Wind instruments include Lip-Driven Brass instruments, Woodwind Reed instruments, Flute and Flue Organ Pipes and Pipe Organs. Examples within each family are given.
  - **Lip-Driven Brass instruments** are for example trumpet and horn
  - **Woodwind Reed instruments** include clarinet, oboe, bassoon and saxophone.
  - **Flute and flue organ Pipes** are based on the principal of jet-resonator interaction with instruments such the Recorder and the Flute.
  - **Pipe Organs** are a sub-class of their own, with acoustics issues dealing with mixtures and mutations, tuning and temperaments, pipe voicing, effect of pipe material, reed pipe ranks, etc.

- **Percussion instruments.** Percussion instruments can be divided into four main families: Drums, Mallet percussion instruments, another type includes cymbals, gongs, plates and steel drums. Finally bells are a sub-class of their own.

- **Drums** are percussion instruments such as kettledrums, timpani, Tom-toms, Snare, most of the Asian percussion, and Tambourines.
- **Mallet percussion instruments** will include the Glockenspiel, the Marimba, the Xylophone, Vibes, Mallets, Chimes, Triangles, and Gamelan instruments.
- **Cymbals, Gongs, plates and Steel Drums** form a sub-class of their own with the addition of Tam-Tams.
- **Bells** include Church Bells, Handbell, Clappers, Ancient Chinese Two-Tone Bells as well as Japanese Temple Bells.

A classical orchestra is a typical ensemble that include most of the above mentioned instruments, placed very specifically depending on their acoustics. Strings are usually in the middle front, followed by the wind instruments with clarinets and flute on the left, brass on the right side, and finally the percussions in the far back.

In this work, the pipe organ will be used as a vehicle for all the investigations because of its importance and relevance to the audio experts and collaborative company Musicom Ltd. This research project has been carried out in close collaboration with two audio experts from Musicom Ltd, one of them is Graham Blyth, a world famous organist who gives regular recitals at Audio Engineering society conventions.

The pipe organ, as pictured in Figure 2.1, is a very large musical instrument. It usually consists of a console (with keyboards and stops as seen in the bottom of the picture), the pipes/windchest (large pipes) and wind supply (a blower / bellows not seen in the picture) which try to generate a constant wind pressure required for the *pipes to sing*.

Pipes can have many shapes, as shown in Figure 2.2 (All pictures from [35]). A Flute, for example, is a rank of broad-scaled pipes, open, covered, chimneyed, or conical, giving a sound similar to a flute or a recorder as shown in Figure 2.2(b). Figure 2.2(c) shows an Oboe which is a reed stop that produces a sound similar to the orchestral

Figure 2.1: Pipe organ (Buffalo, New York)

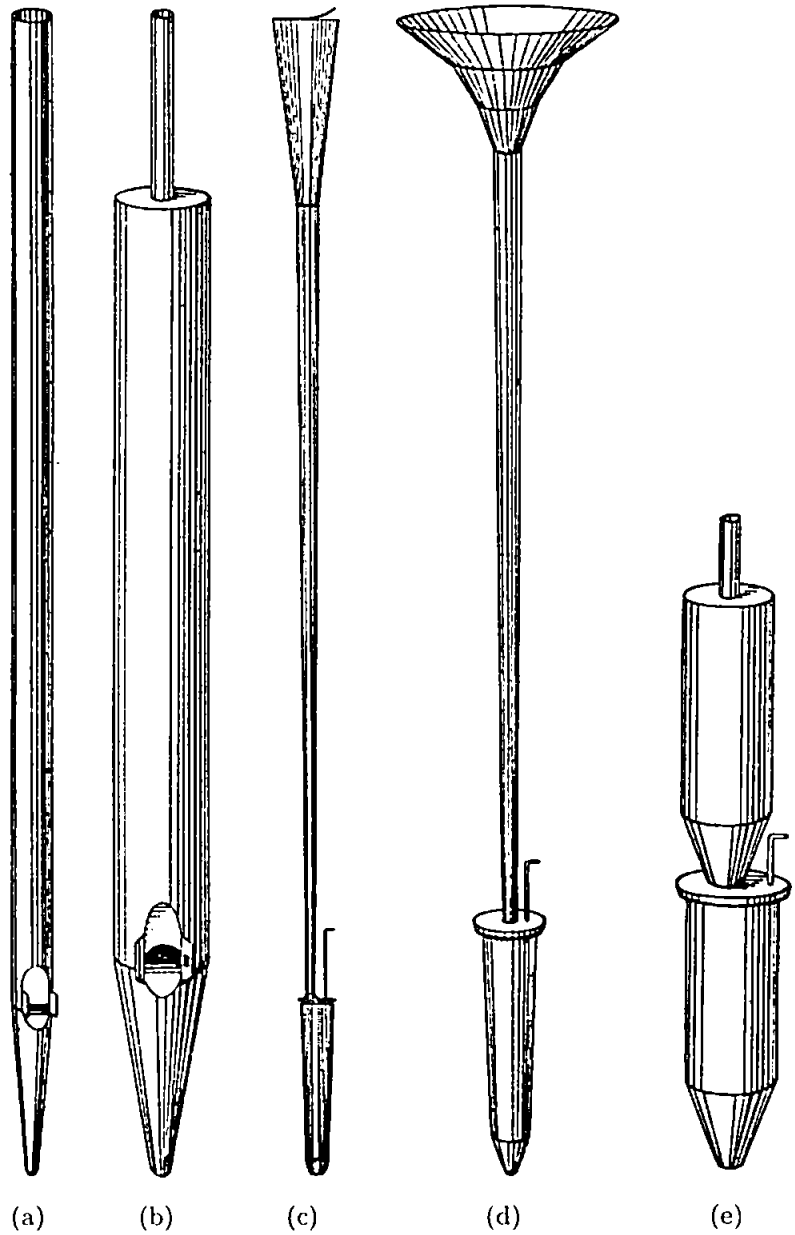


Figure 2.2: Various shapes of pipes: (a) Dulciana, (b) German Flute, (c) Oboe, (d) French Horn, and (e) Chalumeau

instrument. Their shapes can vary from a simple straight pipe, like the Dulciana shown in Figure 2.2(a), to a more complex shape like the French Horn or Chalumeau shown in Figure 2.2(d) and 2.2(e) respectively. Pipes are in majority constructed using materials such as metal and wood. Further detailed information on stops can be found at the

Encyclopedia of Organ Stops website [35].

The pipe organ sound has often been associated with religious music because of its place in church and cathedrals, and also with happy events like wedding ceremonies or sad events like funerals. Due to their size, pipe organs are probably the most complex instrument of all, and often given the name of *the king of instruments*. An example of this gigantism is the world largest pipe organ, the *Wanamaker Organ*[†] installed in the Lord & Taylor Department Store (Philadelphia, Pennsylvania, USA) with a total of 28,482 pipes.

Large and medium size organ pipes are slowly disappearing mainly because of the cost of their maintenance, and often fire is the cause of their disappearance (like the recent example in New York in 2002, which resulted in damages estimated in Millions of US dollars). With their slow extinction, the knowledge used to build them and voice them is also fading away. Interestingly, France, England and Germany have a very large pipe organ heritage, with many famous organ builders still in activity, keeping and pursuing the traditions of pipe organ builders.

To gain detailed information about the pipe organ's long history and complex construction, the book from Hopkins and Rimbault [71] is highly recommended. Acoustics of flutes and flue organ pipes, as well as pipe organs can be found in [39]. Many studies have been published in the Journal of the Acoustical Society of America (JASA) on the acoustics of pipe organs, two recent examples are [94] which details the acoustics of chimney pipes, and [63] with the study of the loudness of pipe organ sounds at different locations in an auditorium. The internet provides good sources of information with the world's most complete encyclopedia of organ stops, the *Encyclopedia of Organ Stops* internet website [35], the mailing-list PipeOrg-L [127] also provides a vast archive of discussions about acoustic and electronic pipe organs from professionals. Finally, Rioux presented his results on sound quality of flue Organ pipes with an interdisciplinary study of the art of voicing in [133].

[†]`http://www.wanamakerorgan.com`

2.3 Sound Analysis Techniques

Sound analysis techniques are methods used to extract important features from signals whether it is to extract an instrument from a piece of music or harmonics from a single sound. Many sound synthesis techniques have been developed each with their respective advantages and limitations. In [28] Ding and Qian presented an interesting method, fitting a waveform by minimizing the energy of the residual. Recently Keiler and Marchand reviewed in [89] techniques to extract sinusoids in sounds using plain FFT, parabolic interpolation parabolic, triangle algorithm, spectral reassignment, derivative algorithm [27] and Phase Vocoder analysis. Two techniques, the Teager Energy operator and Phase Vocoder analysis, are now detailed.

2.3.1 Teager Energy Operator

Kahrs [85] presented audio analysis applications using the Teager Energy Operator (TEO). For discrete-time signal, the Teager Energy Operator, Ψ_d, is defined as:

$$\Psi_d[x(n)] \triangleq x(n)^2 - x(n-1)x(n+1) \tag{2.1}$$

In [177], describing an automated piano modeling method [108] with PEAQ-based optimisation, the use of TEO provided an accurate tracking of both amplitude and frequency of the harmonics of piano sounds. In [62], an automated system is described, in which speech formants and modulations are found using energy separation and TEO.

2.3.2 Phase Vocoder

Phase Vocoder (PVoc) is a well-known digital signal processing technique, its theory has been well described in [129] with its implementation in [130]. It uses frequency domain transformations to implement both analysis and re-synthesis processes.

- **Analysis** - First, the time-domain signal goes through a process of windowing. The signal is sliced in overlapping frames and these windows are multiplied by an analysis window (typically Hanning). The overlapping between two frames is defined by a parameter called the hop size (often 0.5 or 50%). On each frame, a Short-Time Fourier Transform (STFT) is performed to obtain a succession of overlapping spectral frames with minimal side-band effects. The modulus of the STFT results are used to obtain the amplitude of the harmonics, each bin corresponding to one harmonic. Using the measure of the phases between two frames, an *instantaneous frequencies* can be calculated and thus provide the frequency evolutions of the harmonics. Limitations of the Phase Vocoder are the following: 1) the sound is considered to be harmonic and 2) the modulations in the sound should not be too high. Note that both cases can happen with pipe organ sounds. Limitations of the Phase Vocoder have been recently examined in [131].

  Phase-Vocoder outputs are shown in Figure 4.2 for the amplitude evolution of the harmonics and with Figure 4.6 and 4.7 for the frequency evolution of the harmonics.

- **Re-synthesis** - The time domain signal is re-constructed by performing an Inverse Fast Fourier Transform (IFFT) on all frames followed by a successive accumulation of all frames, operation called *overlap-add*.

One particularity of the Phase Vocoder is that it performs well for harmonic signals, however for non-harmonic signals, other techniques have been developed based on the Phase Vocoder.

2.3.3 McAulay-Quatieri

McAulay and Quatieri extended the Phase Vocoder analysis removing its limitation when sounds are non-harmonic and in cases when the modulation is too high. Their work was

based originally on the analysis/synthesis based sinusoidal representation of speech signals [118]. This technique also shares similarities with the speech analysis/synthesis and modification using an analysis-by-synthesis overlap-add sinusoidal modeling by George and Smith [47].

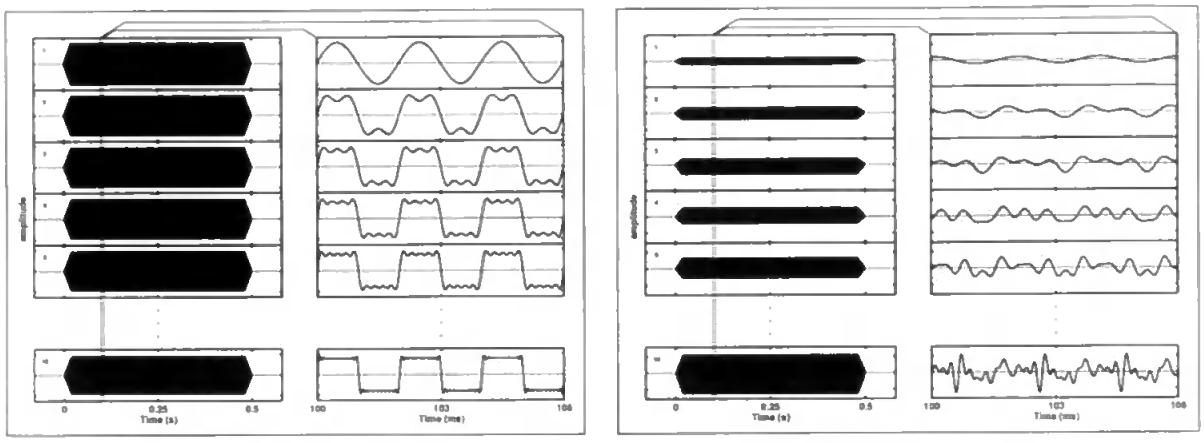
2.4 Sound Synthesis Techniques

Sound synthesis techniques are method used to generate audio signals. Over the last few decades a multitude of sound synthesis techniques have been invented. In the following sections, only a few will be described, they include the additive synthesis, Group Additive Synthesis (GAS), sinusoids plus noise, Frequency Modulation (FM) and physical modeling. A taxonomy of sound synthesis techniques is given in [156].

2.4.1 Additive Synthesis

Using Fourier theory, any signal can be divided into its principal components, in other words sounds can be seen as the summation of sinewave signals each having its own amplitude and frequency. In additive synthesis, each sinewave (also called partial) is modeled by a single sinusoidal oscillator. This also allows specification of individual amplitude and frequency (as well as phase) trajectories over time. The oscillator's output are summed to produce a composite sound waveform.

Recent work by Bertini and his colleagues described some interesting developments of a single VSLI chip that can sustain 1,200 sine-wave oscillators [26] with some management software tool to make most of this sound synthesis power [12]. Additive Synthesis has always been popular for musical applications and its implementation has been discussed for fixed-point DSP in [67] as well as *Single Instruction Multiple Data* (SIMD) and *Very Long Instruction Word* (VLIW) processors in [66]. Figure 2.3 shows waveform examples of additive waveform, with Figure 2.3(a) showing how additive synthesis can approximate



(a) Square waveform approx. (b) Additive Synthesis

Figure 2.3: Additive Synthesis

a square waveform.

2.4.2 Group Additive Synthesis

In [92], Kleczkowski described a method called *Group Additive Synthesis* (GAS). GAS a technique whereby a sound is decomposed into its component sine waves (harmonics), which are then partitioned into groups where they share perceptually identical envelopes. This line of research was followed and improved by Oates and Eaglestone in [123], with results forming the basis of WinPVoc and some of the tools used with the Musicom ADE system.

2.4.3 Sinusoids Plus Noise

Serra and Smith in [145] patented the concept of a musical sound analyzer and synthesizer that uses a model considering sounds with a deterministic component plus a stochastic component. The deterministic component was represented as a series of sinusoids, amplitude and frequency functions for each sinusoid. The stochastic component was represented as a series of magnitude spectral envelopes. Using this representation, sounds could be synthesized and perceptually equal to the original sounds, stored rep-

resentations could also be easily modified in a musical synthesizer for the creation of a wide variety of new sounds.

2.4.4 Frequency Modulation

Chowning was the first to consider Frequency Modulation (FM) for music purposes and patented the technique in the 70's [19]. The most basic FM synthesis consists of a single sinewave modulating a carrier sinewave.

Single-Modulator FM

$$x(t) = w(t)\ sin[2\pi f_c t + I\ sin(2\pi f_m t)] \tag{2.2}$$

with $w(t)$ the carrier amplitude envelope, f_c the carrier frequency, f_m the modulator frequency, and I the modulation index. Figure 2.4(b) shows waveforms examples of FM waveform, generated by a single operator.

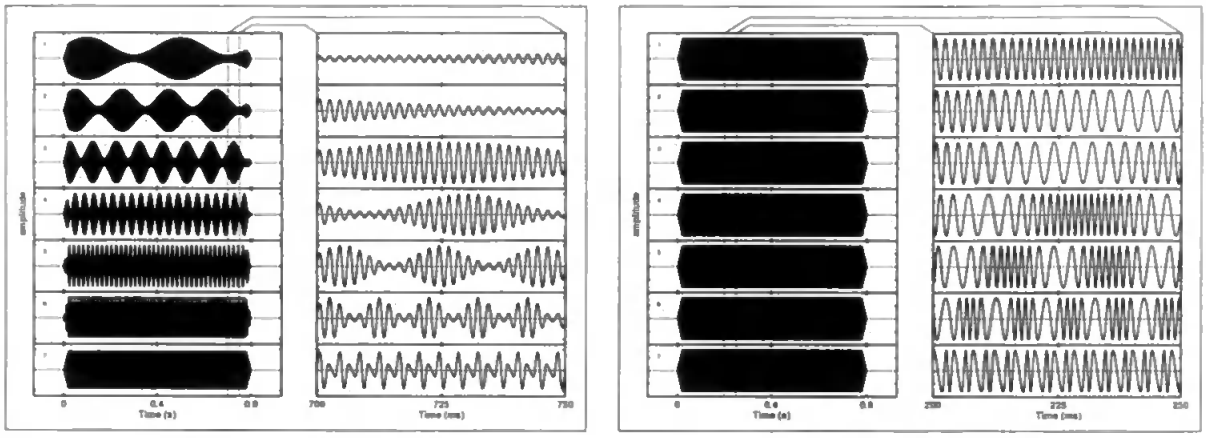
These waveforms are very much different from the amplitude modulation technique, as shown in Figure 2.4(a). Recent sound synthesis models often makes use of both creating more complex sound models [74][9]. Basic FM was extended with more operators as described in [153].

Double-Modulator FM

$$x(t) = A(t)\ sin[I_1\ sin(2\pi f_1 t) + I_2\ sin(2\pi f_2 t)] \tag{2.3}$$

with $A(t)$ amplitude envelope, I_1 and f_1 respectively the modulation index and carrier frequency of the first modulator, I_2 and f_2 for the second modulator.

Recently in [72], Horner presented his research on nested modulator and feedback FM matching of instrument tones, using Genetic Algorithms [68] to optimise the syn-



(a) Amplitude Modulation (b) Frequency Modulation

Figure 2.4: Amplitude and Frequency Modulation

thesis model parameters for instruments such as trumpet and pipa (a classical Chinese instrument). By increasing the number of operators and using them as carriers and modulators, relatively simple structures (also called algorithms) can be created and used to generate more complex sounds. Commercially, Yamaha has developed a variety of FM chips (OPL3) [171] that manufacturers can use in their products. These chips have become very popular since they were introduced in computer sound cards.

2.4.5 Sampling

Sampling is a sound synthesis technique by which one long sample of the sound to be recreated is stored in memory and replayed when triggered, from the key of the piano for example. The basic memory reading process has often post-processing with complex filtering like the famous Morpheus Z-plane of the E-mu Systems [34] which allow smooth morphing between two spectral filter shapes (made of 14 poles). In [93], Kniest and Petersen described a multiple DSP system in which large memory are used to store piano samples with length up to 22 seconds. Rossum detailed many patents, all assigned to E-MU Digital Audio Systems [30], on digital sampling instruments with recent improvement taking advantage of cache memory [140].

Even if sampling is often criticised that it does not re-create all the dynamics of real acoustic sounds, it is still, ironically, the most used synthesis technique in the audio industry recording studios and film industry. Recording studios often collect numerous samplers from manufacturers such as E-MU and Akai [3]. Recently, we have seen many affordable samplers appearing for low budget recording studios.

2.4.6 Wavetable Synthesis

Wavetable is a technique that follows from sampling with the difference that it requires less memory. Basically, it divides a long sample into smaller portions that are read in sequence. A wavetable is a "table with a single wave". The reading process of these tables is defined by the spectral evolution of the sound. This technique has been heavily researched in the late 80's when new hardware could be used to build complex digital electronic oscillators.

Over the years, Horner has published many papers using wavetable to model instruments such as piano [110], Woodstock and Gamelan in [9], as well as Chinese instruments [74]. Zheng and Beauchamp presented an interesting research project with a technique called critical-band group synthesis of piano tones [178]. With this technique, the sound spectrum is divided into 24 different perceptual bands and within each band a perceptual grouping is performed. This technique claims good results for the modeling of a full piano.

Yamaha [173] and Atmel [8] manufacture a large variety of wavetable sound synthesis ICs solutions [7][171]. These chips are used in many products and especially for sounds cards, personal computers, Personal Digital Assistant (PDA) as well as mobile phones.

2.4.7 Digital Waveguide Modeling

Digital Waveguide Modeling [149] has been invented by Julius Smith at the Center for Computer Research in Music and Acoustics (CCRMA), Stanford University. It aims at modeling the acoustics behavior of the acoustics and physics of musical instruments. For example, a piano would be modelled into its components: hammers, strings and a sound board. Smith proposed many improvements for this instrument like the *commuted piano modeling* [150] and patented the technique [148].

Digital waveguide has used for pipe organ from work from Välimäki's doctoral thesis [161]. These techniques have been recognised to be the most realistic sound modeling techniques, indeed they *model*, in the true sense of the term, the real acoustics of the musical instrument. A simple model of an pipe organ has been presented by Zielinski in [179].

Yamaha Corporation seems to have "locked" this technology into a large portfolio of patents, with very few electronic musical instruments available commercially (apart from the Yamaha VL series, see Yamaha VL1-m [172]. In the audio industry, Kunimoto is famous being the main investigator of Yamaha's waveguide developments. All his patents are assigned to Yamaha Corporation (Hamamatsu, Japan) [173]. Toshifumi Kunimoto and his colleagues developments have explored many aspects of digital waveguide modeling for many musical instruments such as brass (e,g, a trumpet) [103], piano [95], and flutes with complex models of conical tubes [115].

In digital waveguide modeling, the estimation [163], optimisation and real-time control of the parameters still remain very challenging. Models often have feedback loops which can make the system unstable, as much as the acoustics of one instrument can be defined (chaotic behavior of air jet for flutes).

Karplus and Strong invented a simple technique to model pluck string using a wavetable filled with noise and looped through a simple feedback filter [88]. This can be assimilated

as the simplest case of waveguide modeling. It produces surprisingly very high quality pluck string sound such as harp and guitar with very little processing power. These techniques are ideal for VLSI[†] implementation [151] and examples of CPLD[‡]/FPGA[§] technology implementation have been described in [10].

2.4.8 Hybrid Synthesis

Often one single sound synthesis technique is not enough for the complexity of certain sounds and instruments, thus hybrid sound synthesis techniques have been developed. The term *hybrid* is used here in the sense of combination. For example, in [9] Ayers and Horner developed some relatively simple models for instruments like Woodstock and Gamelan as well as small Chinese and Tibetan bells [74], Chinese musical instruments such as *Dizi*, *Bawu* and *Sheng* [75]. All these examples use simple amplitude and frequency modulation techniques.

Techniques are usually selected for their respective advantages, whether it is efficiency, quality, control, etc. Yuen and Horner used some hybrid synthesis to synthesise piano tones, a combination of sampling for the attack portion of the sound, cross-faded with classic wavetable synthesis for the sustain part [174]. The majority of the models found in the literature share the common idea of splitting the sound model into two sections: the excitation part and the resonating part. This can be easily related to the physics of musical instruments. Two examples can illustrate this concept: the piano and the pipe organ. In the case of a piano, the string is a resonator and is hit by the hammer, the excitation part. In the case of pipe organ, the pipe is the resonator and the excitation corresponds to the air flow from the wind chamber.

Many Yamaha patents describe hybrid sound synthesis models using for example frequency modulation and digital waveguide models [104], or combination of wavetables

[†]VLSI - Very Large Scale Integration
[‡]CPLD - Complex Programmable Logic Device
[§]FPGA - Field Programmable Gate Array

with digital waveguide model to create an engine exhaust sound synthesizer [105]. Both examples showing interesting combinations of digital waveguide models used as resonators with the other techniques as source signals (i.e. exciter).

Laroche and Meillier described some interesting investigations onto a similar concept with *Multichannel Excitation/Filter modeling* applied for percussive sounds like piano tones [108]. The excitation signal, corresponding to the hammer impact on the string, is generated using time or frequency inverse filtering technique from the analysis of the sounds. No indications are given as to what is the audio quality of the resulting sounds.

2.5 Assessment of Audio Quality

With the recent advances in audio coding [126], there is now a great research interest concerning perceptual issues of audio signal quality. For more details on the principles of perceptual coding, the reader can refer to perceptual filter bank design techniques [14] used in general digital audio compression algorithms [37] and particularly for MP3 and Advanced Audio Coding (AAC) coding techniques [15].

Pioneer research in audio compression by both Fraunhofer[†] and Thomson[‡] has lead to important patents [31] and [53] on audio compression schemes known today as MP3. The recent growth of portable audio devices and mobiles with multimedia functions (audio/video), has emphasised even more the importance of perceived audio quality and techniques to assess it. In the next sections, listening tests, Relative Spectral Error, and Perceived Evaluation of Audio Quality algorithm, known as PEAQ, are described in details.

[†]Fraunhofer Institute for Integrated Circuits IIS, Germany
[‡]Thomson Multimedia, France

Subjective Difference Grade	Meaning
5.0	Imperceptible
4.0	Perceptible but not annoying
3.0	Slightly annoying
2.0	Annoying
1.0	Very annoying

Table 2.1: The ITU-R BS.1116 five grade impairment scale

2.5.1 Listening Tests

The ITU-R standard BS.1116 [80] defines the test procedure for the subjective assessment of high quality audio. The listening tests have to be carried out with the "hidden reference / double blind / triple stimulus" method. In this test method the listeners can switch between the original (reference), A, and two other signals, B and C (triple stimulus). One of the two signal is the processed signal and another original said to be *hidden reference*. Neither the test subject nor the supervisor of the test knows, which of signal B and C (double blind) is the hidden reference, and which one is the processed signal. The listener has to decide which signal is the hidden reference, and judge the overall quality ("basic audio quality") of the other signal. The basic audio quality is measured on the *five grade impairment scale*. This scale covers a continuous range from one to five, in which five levels are defined by a verbal description of the perceived quality see Table 2.1. This scale is similar to the MOS-scale (Mean Opinion Score) used in PESQ algorithm for speech quality assessment (ITU-R P.862) [83].

Listening tests are known to be expensive, time-consuming, requiring specialised sound facilities and a large number of subjects to obtain the required accuracy. Still they are the preferred method of assessing audio quality. The most important issues related to listening tests for audio equipments are detailed in the ITU BS562 recommendation [82]. An important issue is the choice of the audio materials used in these tests.

2.5.2 Relative Spectral Error

To assess the final quality of sound synthesis, listening tests are the traditional method, however to optimise sound synthesis algorithms, other methods have been used. Horner, for example, uses the *Relative Spectral Error* (RSE) [72][169] as a metric to evaluate his sound synthesis algorithms.

$$RSE = \frac{1}{N_{frames}} \sum_{r=1}^{N_{frames}} \sqrt{\frac{\sum_{k=1}^{N_{har}} (b_{k,r} - b^{*}_{k,r})^2}{\sum_{k=1}^{N_{har}} b^{2}_{k,r}}} \tag{2.4}$$

where N_{FRM} is the number of frames (along the time axis) and $b_{k,n}$ and $b^{*}_{k,n}$ is the approximated harmonic amplitude.

Mean Square Error (MSE) and *Power of Signal-to-Noise Ratio* (PSNR) have also been used to measure error. They all have been recognised to be poor at correlating with real perceived audio degradation, and lead to the development of Perceptual Evaluation of Audio Quality (PEAQ) described in the next section.

2.5.3 Perceptual Evaluation of Audio Quality (PEAQ)

The PEAQ algorithm is based on the ITU-R standard BS1387 [81], finalised after many years of independent research on perceptual audio coding by many European research centers [155]. The Opera system is described in [91], and is the only commercially available implementation of PEAQ (See Section 2.5.4).

Figure 2.5 shows the concept diagram of the PEAQ. It is a perceptual-based model, a combination of human perceptual and cognitive models. This technique is objective and closer to the human perceived quality assessment of audio and sounds.

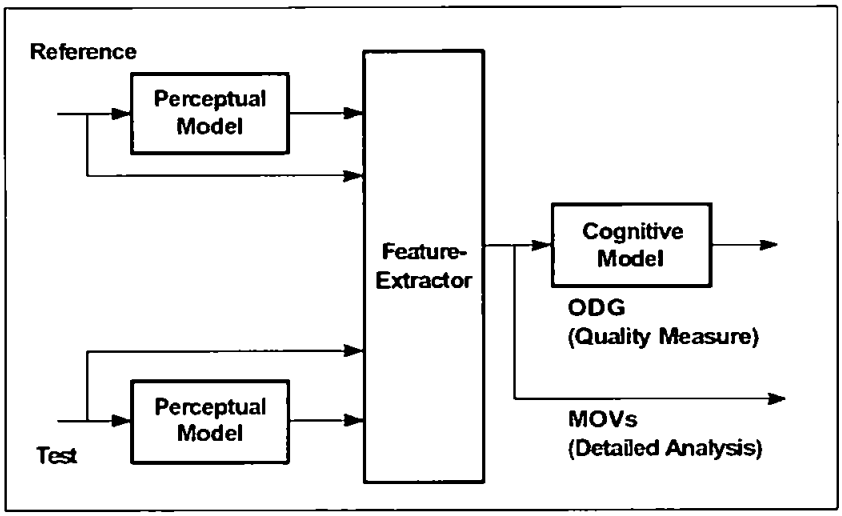


Figure 2.5: Concept diagram of the PEAQ

Perceptual Model

The Perceptual model is a model of the human hearing system. Two sounds, the reference and the test sound, are processed by the two identical perceptual models and the results are fed into a feature extraction module. The feature extraction module produces Model Output Variables (MOVs). The MOVs are the following: **WinModDiff** is the windowed averaged difference in modulation (envelopes) between the Reference Signal and Signal Under Test, **WinModDiff1** is the averaged modulation difference, **WinModDiff2** is the averaged modulation difference with emphasis on introduced modulations and modulation changes where the reference contains little or no modulations, **RmsModDiff** is the Root Mean Squared (RMS) value of the modulation difference, **RmsMissingComponents** is the Rms value of the noise loudness of missing components (used in RmsNoiseLoudAsym), **RmsNoiseLoud** is the Rms value of the averaged moise loudness with emphasis on introduced components, **RmsNoiseLoudAsym** is RmsNoiseLoud plus half the value of RmsMissingComponents, **AvgLinDist** is a measure of the average linear distortions, **BandwidthRef** is the Bandwidth of the Reference Signal, **BandwidthTest** is the Bandwidth of the output signal of the Device Under Test (DUT), **TotNMR** is the logarithm of the averaged Total Noise-to-Mask Ratio, **RelDistFrames** is the Relative

Objective Difference Grade	Meaning
0	Imperceptible
-1	Perceptible
-2	Slightly annoying
-3	Annoying
-4	Very annoying

Table 2.2: Objective Difference Grade (ODG)

fraction of frames for which at least one frequency band contains a significant noise component, **AvgSegmNMR** is the Segmentally Averaged logarithm of the Noise-to-Mask Ratio, **MFPD** is the maximum of the Probability of Detection after low pass filtering, **ADB** is the averaged Distorted Block (i.e. frame), taken as the logarithm of the ratio of the total distortion to the total number of severely distorted frames and finally, **EHS** is the Harmonic structure of the error over time.

Cognitive Model

The cognitive model is a model of the human judgment of sound quality and is based on a mapping between the feature extraction part output (into a final Objective Difference Grade (ODG) and Distortion Index (DI). The ODG values are designed to mimic the listening test ratings obtained from typical test listeners by means of an objective measurement procedure. The grading scale ranges from -4 *very annoying*) to 0 (*imperceptible difference*). Table 2.2 shows the ODG values and their corresponding meaning.

The cognitive model is implemented using a Neural Network [64]. The ITU-R document provides details about its structure and values of its weights [81]. Section 7.5 details future work on PEAQ cognitive model.

There are two versions of PEAQ, a *basic* version, featuring a low complexity approach, and an *advanced* version. The basic model of the PEAQ algorithm is intended for online evaluation of audio quality whereas the advanced model is intended for offline evaluation

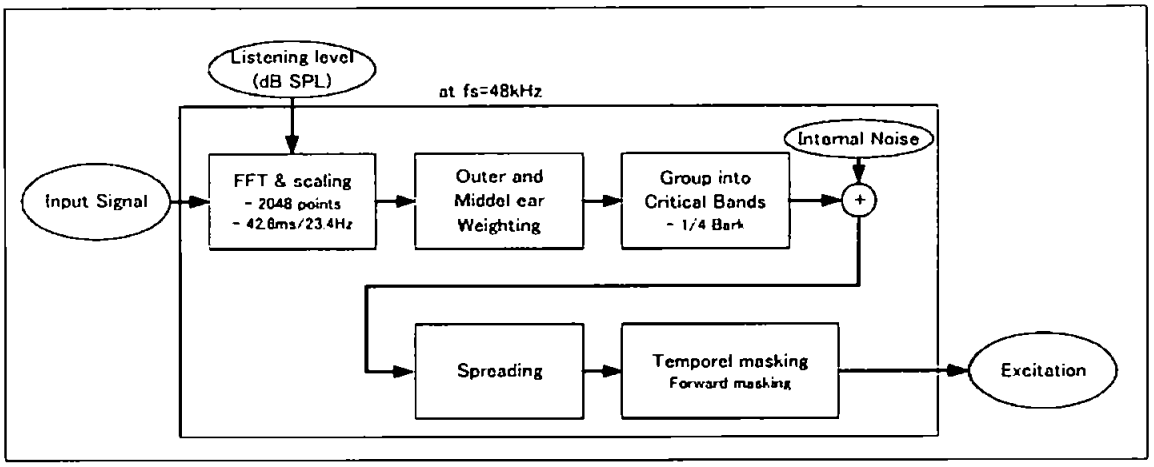


Figure 2.6: PEAQ basic model: FFT-based ear model

of audio quality and requires more processing power. The main difference between the basic and advanced PEAQ models is that the basic model uses a FFT based analysis stage whereas the advanced model uses a bank filter analysis stage.

PEAQ Basic version

The *Basic* version implements an FFT based ear model, as outlined in Figure 2.6 Most features of this model are based on fundamental psychoacoustic principles. Figure 2.6 shows the signal flow from the input signal to the final calculation of the excitation pattern. The processing starts by a transformation of the input signal to the frequency domain. A 2048-point FFT is applied along with subsequent scaling of the spectra, according to the listening level, which has to be input by the user as a parameter. This results in the frequency resolution of approximately 23.4 Hz, and a corresponding temporal resolution of 42.6 ms (at 48 kHz sampling rate).

In the constructive block, the effects of the outer and middle ear are modelled by weighting the spectrum with the appropriate filter functions. Afterwards the spectra are grouped into critical bands, archiving a resolution of 1/4 bark[†] per band. The subsequent adding of *internal noise* is intended to model effects, such as the permanent masking of

[†] $Bark = 13 * atan(0.76 * Freq/1000) + 3.5 * atan((Freq/7500)^2$

Model Output Variable	Purpose
WinModDiff1	Changes in modulation (related to roughness)
AvgModDiff1	...
AvgModDiff2	...
RmsNoiseLoud	Loudness of the distortion
BandwidthRef	Linear distortion (frequency response etc)
BandwidthRef	...
RelDistFrames	Frequency of audible distortions
Total NMR	Noise-to-mask Ratio
MFPD	Detection probability
ADB	...
EHS	Harmonic structure of the error

Table 2.3: Model Output Variables for PEAQ basic version

Model Output Variable	Purpose
RmsModDiff	Changes in modulation (related to roughness)
RmsNoiseLoudAsym	Loudness of the distortion
AvgLinDist	Linear distortions (frequency response, etc.
Segmental NMR	Noise-to-Mask Ratio
EHS	Harmonic structure of the error

Table 2.4: Model Output Variables used in PEAQ advanced version

sounds in our auditory system caused by the streaming of blood and other physiological phenomena. This step is followed by calculation of masking effects. Simultaneous masking is modelled by a frequency and level dependent spreading function. Temporal masking is modelled only partly since the temporal resolution is the same range as the timing of any background masking effects, which therefore cannot be modelled. Nevertheless, experiments have shown that backward masking is very coarsely modelled by side effects of the FFT. Using the feature extractor, 11 MOVs are extracted from the compensation of the ear model output. Table 2.3 lists these 11 MOVs and their associated purpose.

PEAQ Advanced version

The *Advanced* version use some MOVs derived by implementing the ear model of the *Basic* version but in addition to that it introduces a second ear model with improved

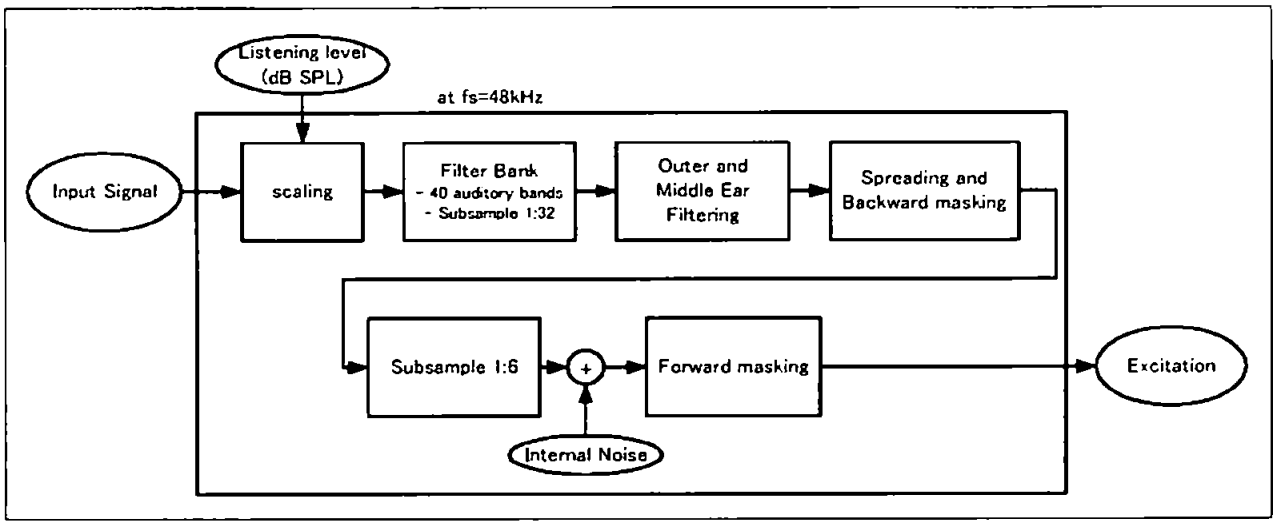


Figure 2.7: PEAQ advanced model - Filter Bank-based ear model

temporal resolution, as illustrated in Figure 2.7. Compared to the *Basic* version, this model performs the time frequency warping using a filter bank, thus grouping the signal into 40 auditory bands with a temporal resolution of approximately 0.66 ms. This allows for a very accurate modeling of backward masking effects. After the calculation of backward and simultaneous masking, the signal is sub-sampled by a factor of 1:6 in order to improve the computational efficiency. After adding the internal noise to the sub-sampled signal and finally modeling the forward masking effects, the output of this model is again the excitation. In comparison to the FFT based *Basic* approach, the temporal resolution is improved, thus allowing for better simulation of temporal effects, at the cost of frequency resolution and computational complexity. Due to the combination of parameters derived from both of the ear models, the number of MOVs used by the *Advanced* version to derive the final quality measure could be reduced to five. The MOVs used by the *Advanced* version are listed in Table 2.4.

Objective Difference Grade

The Objective Difference Grade (ODG) is the output value of PEAQ method that corresponds to the Subjective Difference Grade (SDG) in the subjective domain. The reso-

lution of the ODG is limited to one decimal (e.g. -1.9). However, one should be cautious and not generally expect that a difference between any pair of ODGs of tenth of a grade is significant. The same remark is valid when looking at results from a subjective listening test. The ODG can also assume positive values, as for example, such values can occur because PEAQ use the cognitive model to map the MOVs to the results of subjective listening test.

In the case of subjective listening tests, the SDG can assume a positive value, when a test person has incorrectly assigned the reference and test signal. The Distortion Index (DI) has the same meaning as the ODG. However, DI and ODG can only be compared quantitatively but not qualitatively. DI is characterized by a saturation that is less than the saturation of the ODG value. Furthermore, the range of values is different. As a general rule, ODG should be used as the quality measure for ODG values greater than approximately -3.6. The ODG correlates very well with subjective assessment in this range. When ODG value is less than -3.6 then DI should be used.

PEAQ Audio Materials

The PEAQ algorithm has been tested using a set of audio materials selected by audio experts. These experts are professionals from the audio industry and research academics. The list of the test signals include items such as sound from castanets, clarinet, Claves, Flute, Glockenspiel, harpsichord, Kettle drum, Marimba, Piano Schubert, Pitch pipe, Ry Cooder (clavechord type of music), Saxophone, Bag pipe, speech from a female (in English), speech from a male (English), speech from a male (in German), Snare drums, Soprano Mozart, Tamborine, Trumpet, Triangle, Tuba, extract of Susanne Vega (from Tom's Diner) and finally a Xylophone sound.

Each item can have some specific artefact associated with it, such as: 1) transients effects like pre-echo sensitive, smearing of noise in temporal domain, 2) tonal structure noise sensitive and roughness, 3) natural speech (critical combination of tonal parts and

attacks): distortions sensitive, smearing of attacks, 4) complex sound: stresses the DUT, and finally 5) high bandwidth, thus possible stresses the DUT, loss of high frequency, programme-modulated high frequency noise.

Each test signal has a duration of 10 to 20 seconds, however, it is likely that the critical part of the test signal, which unveils most of the artefacts, it limited to only a short part of the duration. These perceptual artefacts, specific to audio coding compression algorithms, have been discussed in Erne's paper [38] associated with the CDROM created by the AES Technical Committee on Audio Coding.

PEAQ Performance

In order to validate the performance of PEAQ model, a number of different criteria may be relevant. The correlation between ODG and SDG has been chosen as an obvious criterion for its evaluation. In addition, two further criteria that consider the reliability of the mean value were used for validation: the Absolute Error Score (AES) and the Tolerance Scheme. The validation tests performed by ITU-R showed that PEAQ predicts the perceived quality with high-accuracy and was superior to any previously existing measurements method. Further detailed information can be found in the ITU-R recommendation BS.1387 document [81] and [155].

2.5.4 OPERA^{TM} Voice/Audio Quality Analyzer

The PEAQ algorithm is a standard and thus has been ratified by world experts on audio quality assessment. The only commercial version of PEAQ is provided by Opticom GmbH[†] [91] as the OPERA^{TM} Voice/Audio Quality Analyzer. OPERA^{TM} is based on a Dual-Pentium III machine running Microsoft NT 4.0 operating system, built with industry standard and features professional analog and digital audio interfaces. Figure 2.8

[†]Opticom GmbH, Germany

Figure 2.8: OPERA™ Voice/Audio Quality Analyzer (Opticom GmbH)

shows a picture of the OPERA™ Voice/Audio Quality Analyzer used in this research work.

2.6 Summary

In this chapter, a short introduction on acoustic musical instruments was given with detail on the pipe organ used as the main vehicle for this research work. Sound analysis techniques have been presented with emphasis on Phase Vocoder, technique used in the sound analysis engine of the intelligent audio system described in Chapter 4. Sound techniques have been presented in Section 2.4. Finally, methods for the assessment of audio quality have been detailed with the listening tests, Relative Spectral Error, and

the Perceptual Evaluation of Audio Quality (PEAQ) algorithm based on the the ITU-R BS1387.

Chapter 3

Artificial Intelligence approach to Sound Synthesis Parameters Optimisation

3.1 Introduction

This chapter presents a novel approach to sound synthesis parameter optimisation based on Artificial Intelligence. The novel approach has been developed and implemented in software as an automated sound design system based on fuzzy model of audio expertise. This audio expertise was captured from two audio experts and exploited to help design sound, mimicking how the audio experts do. This chapter gives an overall description of the intelligent audio system. This approach, presented for the first time at the Audio Engineering Society convention in New York, December 2001, appeared to be a first in computer music.

Figure 3.1 shows a conceptual diagram of the intelligent audio system developed in this thesis for sound design and modeling musical instruments. The intelligent audio system consists of four main blocks: a Sound Analysis Engine (1), an Audio Feature Processing Engine (2), a Sound Synthesis Engine, and finally (3) an Audio Quality Assessment Engine (4). Each block will be briefly described in the following sections and in more details in the next chapters. To complete the description of the conceptual diagram, the

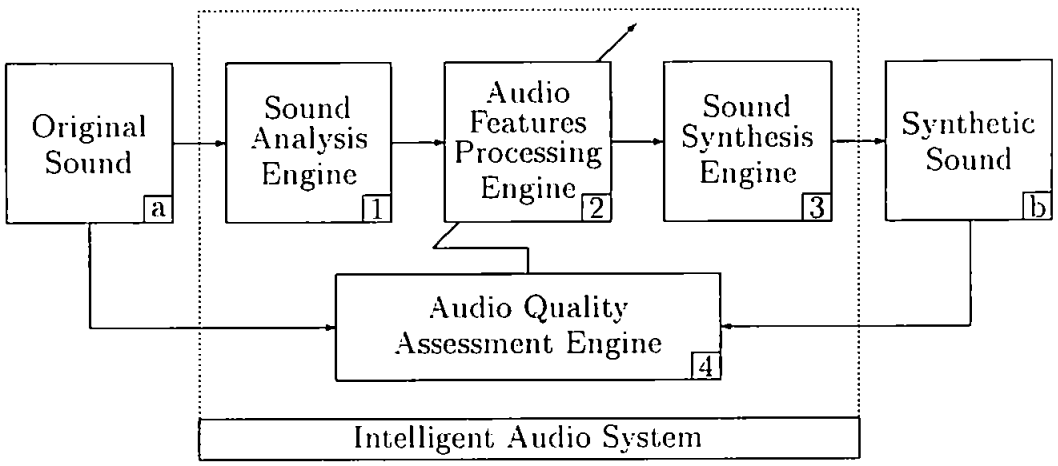


Figure 3.1: Conceptual diagram of Intelligent Audio System

input of the system is an original sound (a) and the intelligent audio system produces a synthetic sound (b). The performance evaluation is based on the audio quality assessment engine which compares the original and synthetic sounds to obtain an objective sound quality index. Examples of the results from experiments carried out on a large pipe organ sound database are presented in Chapter 6. This sound quality index is used in a feedback control loop, in an attempt to optimise and fine-tune the audio features processing engine parameters.

3.2 Sound Analysis Engine

As shown in Figure 3.1, the Sound Analysis Engine (block 1) serves as a front-end of the intelligent audio system and is shown in more detail in Figure 3.2. The development of the intelligent audio system has been based on sound database using CDROMs provided by Musicom Ltd. A complete list is given in Appendix A.1. This large (1.9 GBytes) sound database consists of WAVE files in stereo but in a special format. The two channels are as follows: the left channel is a recording very close to the mouth of the organ

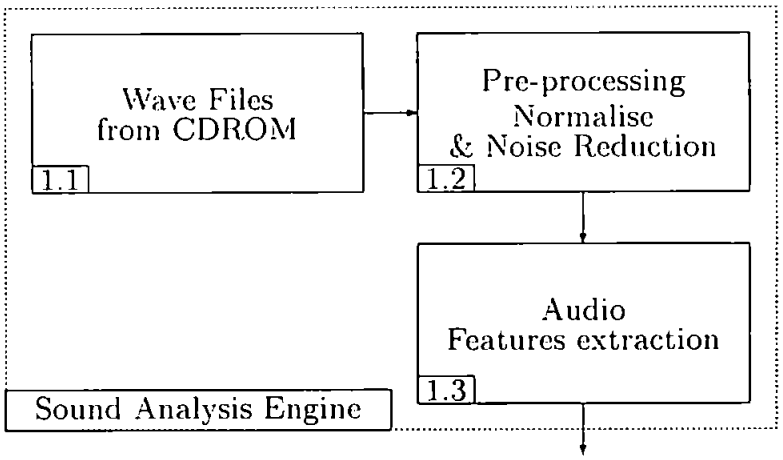


Figure 3.2: Diagram of the Sound Analysis Engine

pipe and the right channel is recorded near the extremity of the pipe [97]. A Sound Analysis DialogDox (shown in Figure 3.15 at the end of this Chapter) is presented to the user (i.e. audio experts) after loading the sound file. This DialogBox allows the user to audition the sound, and to adjust the fundamental frequency for analysis (presently estimated using auto-correlation [78]) for fine tuning, or increase / decrease it by musical steps. Adjustments automatically re-calculate and update the corresponding number of bands of the Phase Vocoder. The WinPVoc implementation of the Phase Vocoder uses a Hanning window. Different types of windows have been added (Hamming, Chebyshev with 120dB relative sidelobe attenuation, and Blackman) that can be selected from the Preferences DialogBox, their use will alter the sound slightly.

The audio expert sometimes manually adjusts the fundamental analysis frequency of the sound analysis engine using experience (name of the sound, type of the pipe, tuning expertise, etc.). The maximum number of harmonics, N, the Phase Vocoder analysis can extract from the sound can be calculated as:

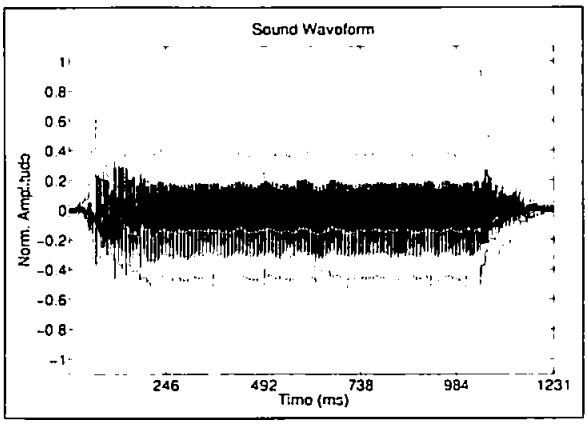
$$N = \lfloor \frac{F_s}{F_o} \rfloor \tag{3.1}$$

where F_o is the center frequency of the low pass filter centered on the estimated funda-

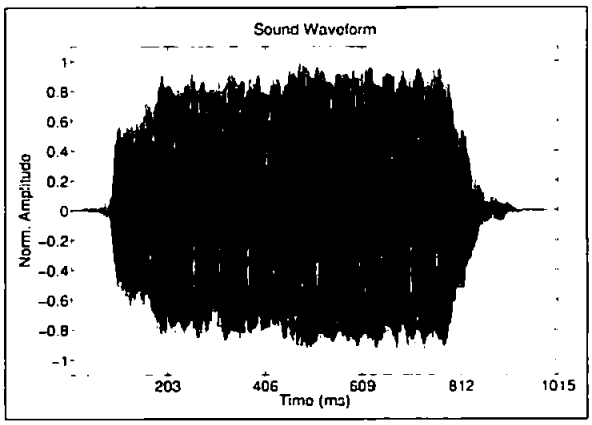
mental frequency of the sound, F_s is the sampling frequency of the sound recording, and $\lfloor\ \rfloor$ means floor function (i.e. rounded to the nearest integer towards minus infinity). All the wave files manipulated in this project have a sampling frequency of 48 kHz (corresponding to professional audio industry standard and preferred for sound files to be used with the PEAQ algorithm [81]).

The user can then launch the sound analysis process (which can also can be automated from the preferences). A table with the Musicom notation (wave filename) corresponding to musical keyboard notes and frequencies is given in Appendix A.1. A quick audition reveal that sounds occasionally require some pre-processing. This can also be automatically detected by the sound analysis engine. Pre-processing includes normalisation and background noise reduction often due to difficult recording conditions (e.g. church and cathedrals). The recording process, carried out by the audio experts, uses multi-position close-miking techniques to minimise, as much as possible, the acoustic background noise on the recordings. The audio expert admitted that it was still an area in which there is still need to improve collecting high quality recordings.

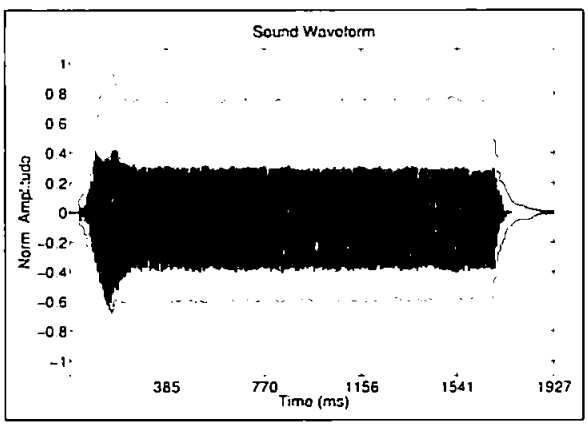
The estimated fundamental frequency of the sound is used as main the parameter for the Phase Vocoder analysis which computes the amplitude, frequency and phase envelopes (trajectories). Background theory on Phase Vocoder analysis was given in Section 2.3.2. Examples are presented in Figure 3.4 and Figure 3.5. It is worth noticing that audio experts prefer to visualise the time-varying spectral structure of the sound using logarithmic scale (following the human perception scale [180]), as it expands small variations, and shows the amplitude difference between harmonics much better. Figure 3.5(a) and Figure 3.5(b) show respectively the linear and logarithmic scales. Notice that software visualisation features allow to zoom in/out and rotate the figure for closer inspection of the time-varying spectral structure.



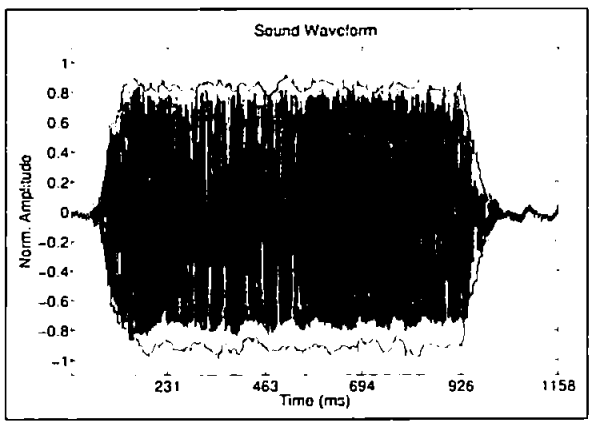
(a) Basson



(b) Flute



(c) Horn



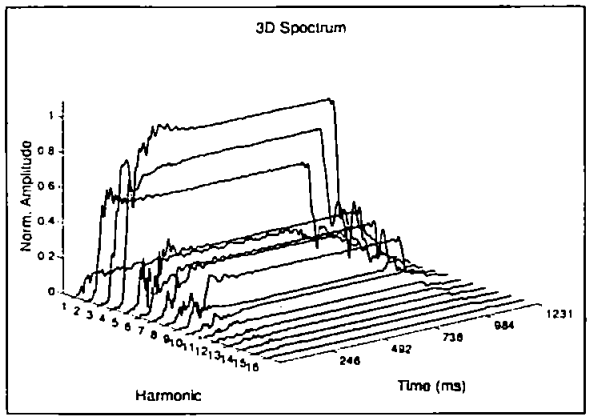
(d) Principal

Figure 3.3: Waveforms of pipe organ sounds

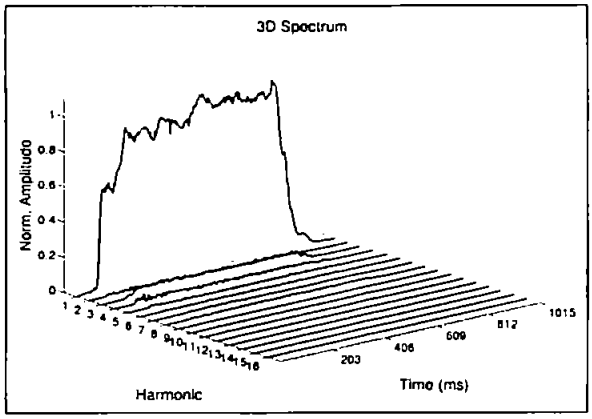
3.2.1 Pre-processing

As stated above certain sounds required pre-processing. Stereo sounds are first converted into mono. The current system deals with sounds in stereo or multi channels format, by user selection of one channel, however future developments can easy accommodate multi channels formats. The sound analysis engine then proceed with some normalisation of the sound level to a full maximum.

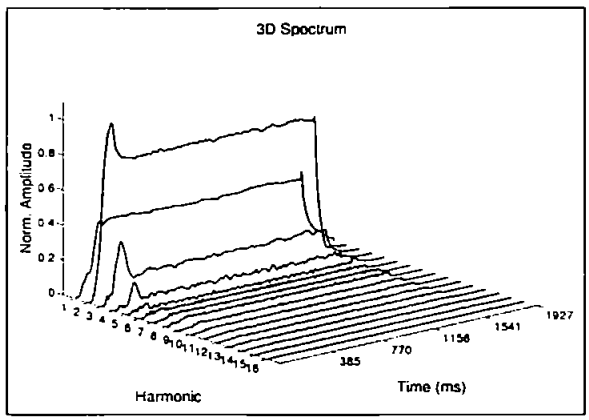
The pre-processing was left outside the main focus of this research project, and work with "clean" sounds was often preferred. One important issue with noise reduction process is that pipe organ sounds are very breathy and it is often difficult to distinguish accurately what is background noise of what is part of the sound itself. This has often



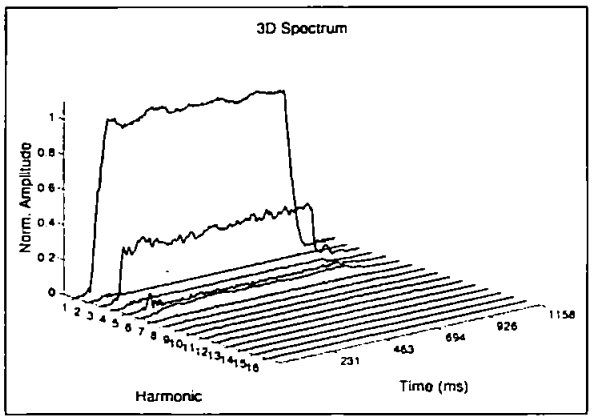
(a) Basson



(b) Flute

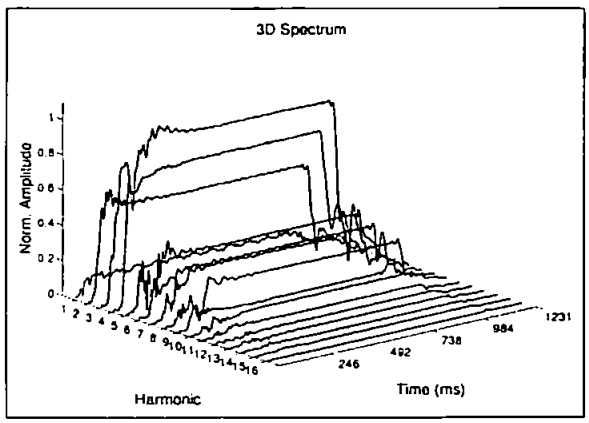


(c) Horn

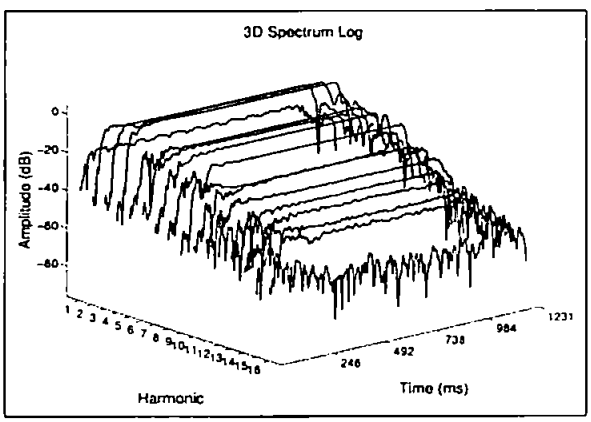


(d) Principal

Figure 3.4: Spectrum Lines of pipe organ sounds



(a) Linear scale



(b) Logarithmic scale

Figure 3.5: Spectrum Lines with Linear/Logarithmic scale (Basson sound)

been left to the audio experts to deal with as they have extensive experience with pipe organs sounds recorded in complex acoustics such as churches or cathedrals. There are many software tools for noise reduction particularly effective for hiss removal often requiring a spectral fingerprint of the noise to be removed. Noise reduction experiments were carried out on a small subset of the sound database using spectral subtraction noise reduction technique. However, results have not been considered satisfactory from both the author's and the audio experts' opinions. More work will be required in this direction. Future work will investigate statistical (Bayesian) model-based approach noise reduction techniques [48] and audio signal enhancement based on psychoacoustic principles [167]. This will be combined with an "Advanced Sound Database" described in Section 7.5.2).

3.2.2 Audio Features

The sound analysis engine extracts audio features from time-varying harmonic components. Audio features can be divided into two categories temporal features and spectral features. The temporal features are time related evolution of the sound's overall amplitude envelope, whereas spectral features are related to time-varying evolution of the spectral envelopes, i.e. each harmonic [117]. The list of audio features consists of the amplitude envelopes, the frequency envelopes, the harmonic distribution, the modulation of the amplitude envelopes, the modulation of the frequency envelopes, the pitch, the brightness and noise factor. A detailed description of each feature is given in Section 4.3.

3.2.3 Audio Features Extraction

In the following sections, methods used to extract the audio features will be detailed. To extract temporal features of the sound (attack/decay/release time segmentation), a split-point estimation technique [84] was used, that involved smoothing with a gaussian envelope using convolution a Digital Signal Processing (DSP) technique [78]). Using time

derivatives of this smoothed envelope, maximum and minimum variations can be found and thus obtain each of the five segments. Each split-point has a variable amplitude (in percentage of the maximum amplitude of the partial) and time. The shape of each segment can be exponential, linear or logarithmic. This method is proven to be more stable than the conventional percent method in which percentage threshold of the envelope is used to the estimate the split-point values.

A simpler method than the originally used in [60] was found by calculating the histogram of the amplitude envelope and finding its maximum. This value corresponds to the sustain level (most of the values of the amplitude envelope that ends in the same histogram bin). The attack time is estimated using a convex hull polygon that encapsulates the envelope. An example is shown in Figure 3.6 in which the sustain level is a horizontal dashed line and estimated attack time shown as a vertical thick line. This simple technique was found to be very reliable for the estimation of the sustain level and attack time. It is worth noticing that the audio expert often augment the attack time slightly to avoid transient audible artefacts at the cross point between attack and sustain. More examples are shown in Section 4.3.1 with Figures 4.1(a) to Figures 4.1(f).

The audio features, as described in [60], are 7 main spectral features: the energy of the fundamental (1^{st} modified Tristimulus), the energy of harmonics 2 to 4 (corresponding to the 2^{nd} modified Tristimulus), the energy of the remaining harmonics (3^{rd} modified Tristimulus), the frequency of the fundamental, the spectral centroid, the ratio of the energy of even harmonics to the total energy of the signal, and the ratio of the energy of odd harmonics to the total energy of the signal excluding fundamental as described in [102] and in [101]. These are called modified tristimulus and were first introduced in [128] as a timbre equivalent to the color attributes in computer vision research. More details together with examples are given in Section 4.3.4.

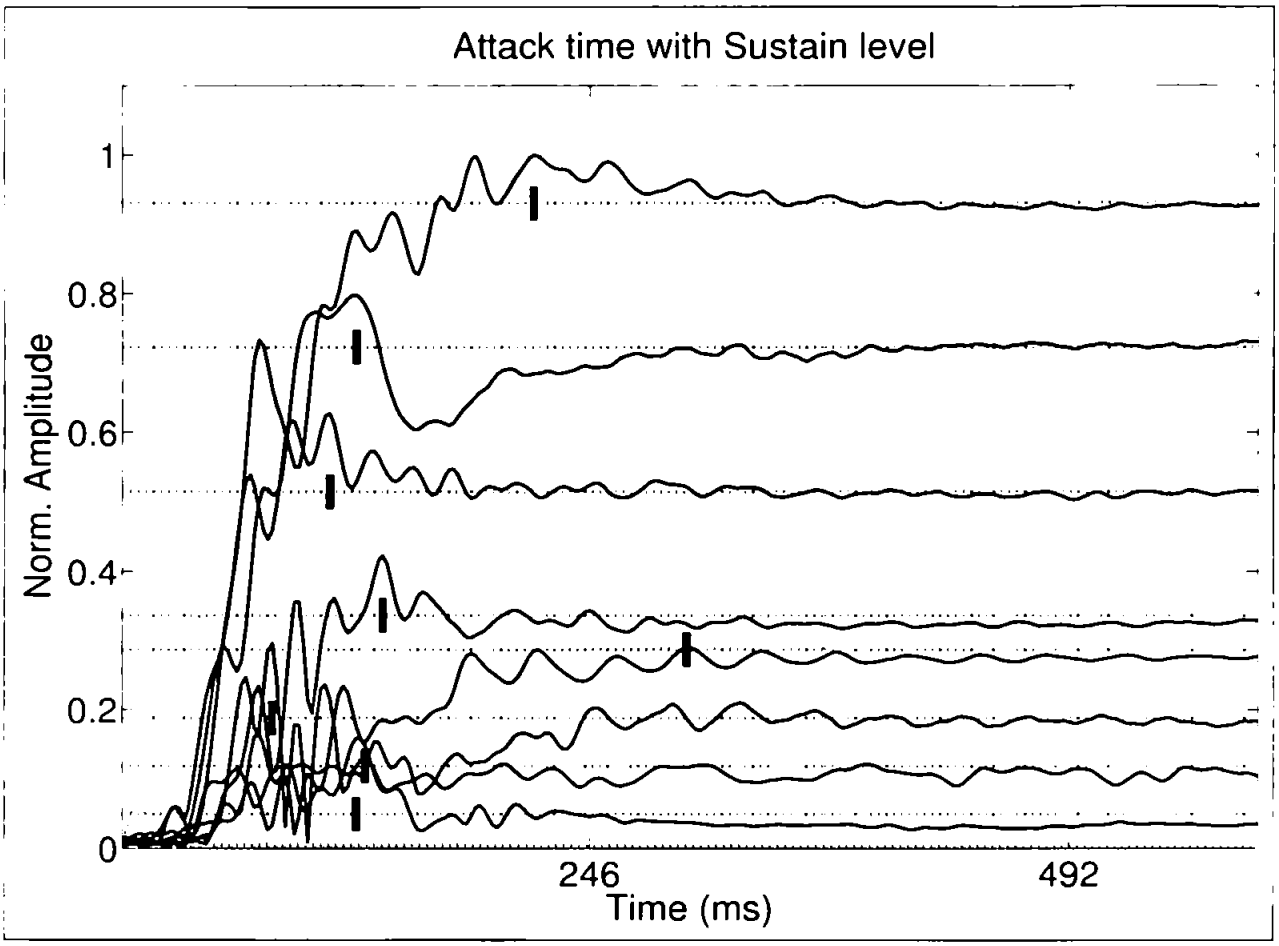


Figure 3.6: Estimation of attack time and sustain level

3.2.4 Descriptive Report

The sound analysis process offers the audio experts the opportunity to give lots of comments on the characteristic of the pipe organ sound. Thus, the system tries to generate a similar descriptive report with in human terms related to pipe organ builders. an attempt to give a musical type of description of the sound using terms taken from the pipe organ vocabulary. Table 3.1 presents series of linguistic terms developed in collaboration with the audio experts. These linguistic terms are very typical of the *jargon* used by the audio experts.

Examples of such descriptions were given by the audio experts [97] and helped for creating these reports. **Flutey** - A *flutey* sound would be a soft sound with predominant

first harmonic and a limited range of about 10 harmonics decreasing rapidly in amplitude beyond the 5^{th}. **Bright** - A *bright* sound is a clear sound with harmonics that extend into the upper frequencies of about 12-15 kHz. **Slow** - A *slow* sound has low frequency note with harmonics that take longer than 50 ms to start. **Nasal** - Typically *nasal* sounds have louder harmonics in the 3^{rd} to 9^{th} range than at the 1st harmonic. **Breathy** - A *breathy* sound which has air noise that is about the same amplitude as the harmonics. This can also be a sound which has a very unstable upper order harmonic structure. The sound is unstable in terms of pitch or amplitude. **Harsh** - A sound is said to be *harsh* when typically the harmonic amplitudes stay the same right up to say the 50^{th} harmonic, with very little variation in the their level. The 1^{st} to 5^{th} harmonics may be lower in amplitude than the others. **Untidy** - If the variations of the harmonics in sustain part are large then the sound is said to be *untidy*, **Even** - when the variations of the harmonics in sustain part are medium then the sound is said to be *even*, **Percussive** - A sound is often said to be *percussive* if the harmonics in the attack part are "short".

It has been found difficult to develop an accurate definition of each term and a sound analysis of the full sound database [57] is helping toward this direction. The motivation is to try to correlate combinations of specific audio features to these linguistic terms using statistical techniques such as Multidimensional Scaling (MDS). Pioneer research on timbre characterisation was carried out by Grey [51]. Recent advancements mainly focus on musical instruments identification [101][114]. Work on pipe organ semantics has recently been complete by Rioux focusing on the process of voicing a flue organ pipe [134][29]. In other research domains such as audio/speech quality for mobile phone, similar investigations have also been recently published [116].

Description	Terms used by pipe organ builders
General impressions	old, noisy, pleasant, relaxed, simple, stable, strong, tensed, thin, undefined, unfocused, unpleasant, unstable, warm, weak
Transient part	aggressive, strong, weak sounds like chiff, sounds like cough, sound like hiss, fast, gentle, long, short, slow, soft, connected, disconnected, integrated, related
Steady state	airy, breathy, bright, clean, clear, cold, dirty, dull, floppy, flowy, fluffy, flutey, free, full, harsh, horn-like, leaky, loose, nasal, oppressive, reedy, rough, round, sandy, sharp, singing, splitting, stringy, thin

Table 3.1: Linguistic descriptors used by the pipe organ audio experts

3.3 Audio Features Processing

The audio features engine of the intelligent audio system is main contribution of this research project, namely the fuzzy model of audio expertise described in Chapter 5. This Artificial Intelligence-based model is based on experience gained over many years of work in the field of electronic pipe organ, and synthesis of pipe organ sounds. To develop such a complex model, it was necessary to learn first about all aspects of sound design. To achieve this, software developments have been made to the tool WinPVoc and are described in the next sections. These software features were important to 1) automate time-consuming task part of the sound design process to reduce the audio expert's workload and at the same time 2) help the author to gain better understanding of the psychoacoustic and engineering issues sound analysis and sound synthesis. These developments include manual clustering (Section 3.3.1) a feature added to help the audio experts to investigate the perceptual effect of each harmonic in the clustering and re-synthesis processes. Algorithmic methods have been implemented with brute force enumeration, and limited permutation of the first 16 highest harmonics (Section 3.3.2). A perceptual masking (Section 3.3.3) process [106] has been implemented to assess the potential masking effect of certain dominant harmonics onto others. Spectro-temporal simplifications (Section 3.3.4) have been also implemented to assess the perceptual effect of data reduction on

amplitude and frequency envelopes similar to the work of McAdams [117]. The Agglomerative Hierarchical Clustering (AHC) algorithm of WinPVoc have been improved with the addition of distance metrics (Section 3.3.5). Finally, more advanced clustering algorithms such as K-Means Clustering (KMC), the Linde-Buzo-Gray K-Means Clustering algorithm (KBG-KMC), and Fuzzy C-Means Clustering (FCMC) have been implemented in MATLAB with preliminary results reported in [56].

3.3.1 Manual Clustering

To learn about the audio expert's work and to try to capture and exploit this knowledge in a computational model, a harmonic toolbar (docked on the right side of the Harmonic window, with *toggle mode* buttons) was added to WinPVoc to allow manual selection of the harmonics. The audio expert has now the possibility to mute individually any harmonics that are used by the re-synthesis engine. This was made only to the first 24 harmonics, with another button that mutes the remaining harmonics from the 25^{th} to the last one (corresponding to the number of bands of the Phase Vocoder analysis). An additional *mute all* button allows to "start from scratch", and gradually adding harmonics and assess their perceptual weight and impact on the final sound quality.

The access and manipulation of the harmonics creates a log-file that keeps track of all the actions of the audio experts. It was hoped that the study of these log-files would give clues and better understanding about the sound design process, and would also help to try correlate some of the the audio experts work method with psychoacoustic knowledge [180]. Two points to be made:

- The harmonic toolbar and manual clustering feature helps greatly the audio experts, giving freedom to experiment on sound design with individual harmonics. Most importantly, it allows them to assess the perceptual contribution of each individual harmonic on the final sound quality.

- For the author, it was found rather difficult to try to interpret the log-files. Only the most simple cases could be easily re-created. Many demonstrations by the audio experts proved to be give a much clearer understanding of the sound design process, with the true explanations about the harmonics manipulation during the sound design.

3.3.2 Ranking and Permutation

A more computational approach was then taken to automate as much as possible time-consuming tasks of the audio experts day-to-day work. This was also helping to consolidate the understanding of perceptual clustering, basis of the audio expertise. Many variants of ranking and permutation algorithms have been developed to enumerate a limited number of harmonic combinations. Only the first (highest) 24 harmonics were considered. Audio experts were also very keen to investigate the effect of combining certain harmonics and assess their perceptual weight into the final sound quality.

Experiments with ranking (sorting) and combination algorithms were performed in which harmonics amplitude data (evolution of the amplitudes over time) are ranked following their maximal and sustain values, and a subset was used to re-generate a sound. One should note that these experiments did not consider any perceptual issues about the harmonics. Perceptual masking was considered separately.

In terms of processing requirements, modern computers can easily generate and process GBytes of sound data and this was not found as a limitation. The specifications of the computer used for this research project are Pentium IV 2.2 GHz processor and 1 GBytes of memory. Processing time to generate ranking and permutations data, followed by the sound re-synthesis with final sound quality assessment, is in the order of minutes. This only depends on the sound length and its pitch, as the lower the pitch the higher the number of bands in the Phase Vocoder analysis and thus creating larger amount of data at the analysis stage. Timing functions, based on the computer internal clock, have been

added to give indications about the processing time required by the analysis, clustering and synthesis tasks. In summary:

- Modern computers can easily sustain heavy and time-consuming audio processing tasks of large amount of sound data. This should be used as much as possible to reduce the audio experts work-load from basic and repetitive tasks.

- There was a need for the most time-consuming tasks of the sound design process to be fully automated. The use of scripts and shell environment such as like MATLAB, facilitates this automation. Research from Horner using small combinational enumeration of harmonics to help his work based on Genetic Algorithms [73](publication which later procured interesting comments by Bristow-Johnson in [18]) demonstrated the potential of nature-inspired search for synthesis optimal parameters, using computer for intensive processing tasks.

- The timing functions added to WinPVoc are helping 1) to measure the time taken by analysis, clustering and synthesis task for individual sound and help to estimate the time that would be required for a large sound database and 2) to benchmark FFT and convolution routines for future development of WinPVoc using recent optimised versions of FFTW [40].

3.3.3 Perceptual Masking

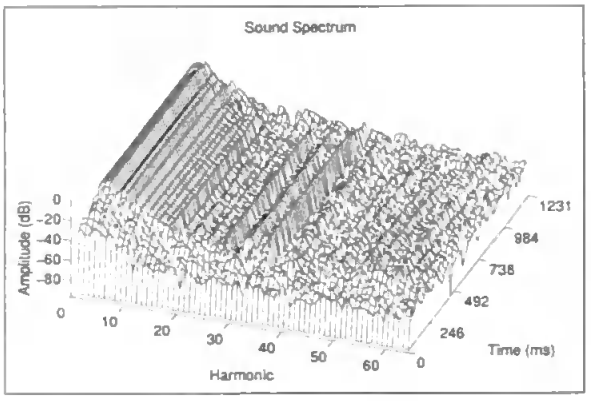
A perceptual mask was calculated on a sound basis, to assess if certain dominant harmonics mask others. Few examples are shown in Figure 3.7 for Basson and Horn sounds, and Figure 3.8 for Flute and Principal sounds. Figure 3.7(e) and 3.8(f) show clearly horizontal troughs in the difference spectrum (difference between original and masked one) showing that certain harmonics can be potentially masked by others (in-depth discussion on perceptual masking can be found in [180]). Perceptual masking is an important aspect

of sound synthesis as the final judge is the human ear and thus perceptual masking can be seen as form of optimisation. We draw some small conclusions:

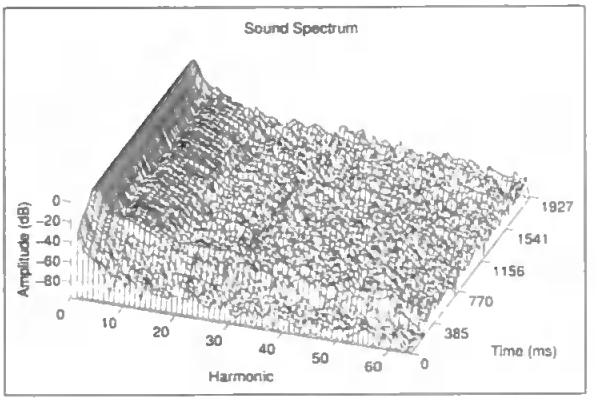
- Perceptual masking is a very processing intensive. Visual inspection of the mask difference and the harmonic distribution provide useful clues about potential masking effect on harmonics.

- In most cases, the perceptual masking for a single sound is not considered to be too obvious. However in the context of musical performance, in which the total number of harmonics used and added together to create the final music will certainly benefit more from the perceptual masking effect.

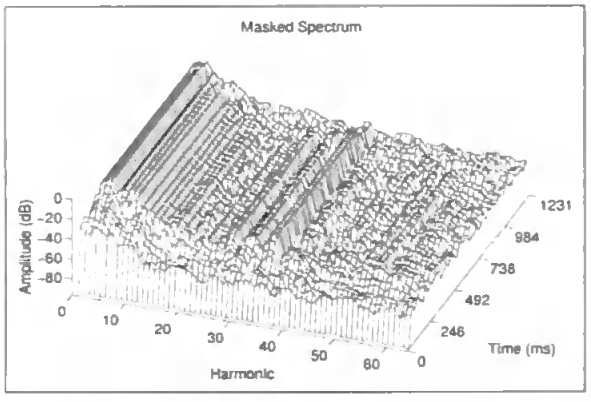
- This perceptual masking issue should be considered when looking at the full modeling of a musical instrument and at the design stage of future Musicom ADE evolutions. Ideally, one could imagine the development of DSP chip that would fit between the control part and sound synthesis DSP of a electronic musical instrument. It would monitor the control signals and optimise the use of synthesis DSP resources using the concept of perceptual mask.



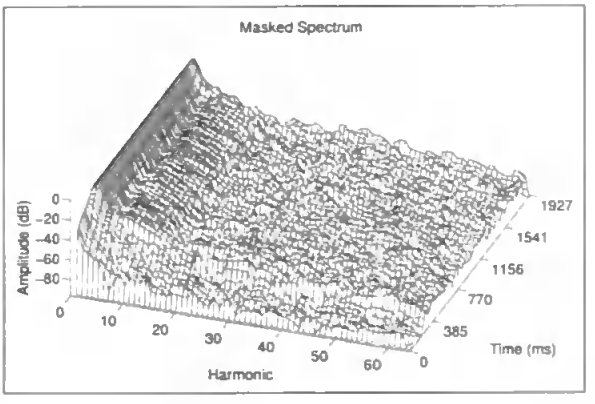
(a) Original spectrum



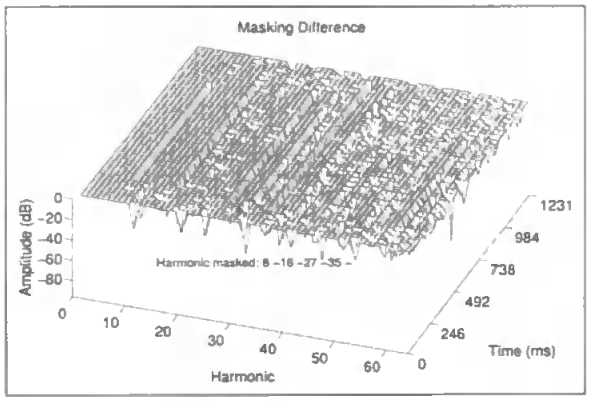
(b) Original spectrum



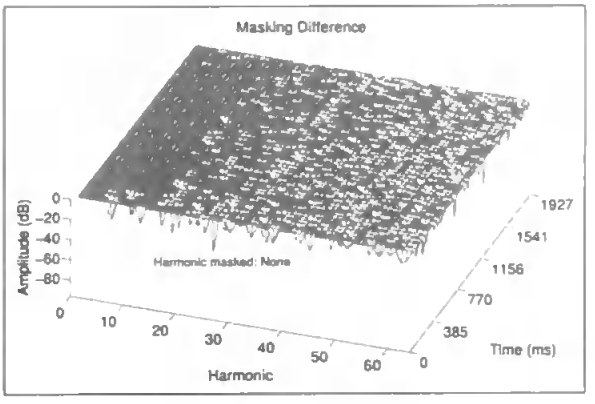
(c) Masked spectrum



(d) Masked spectrum

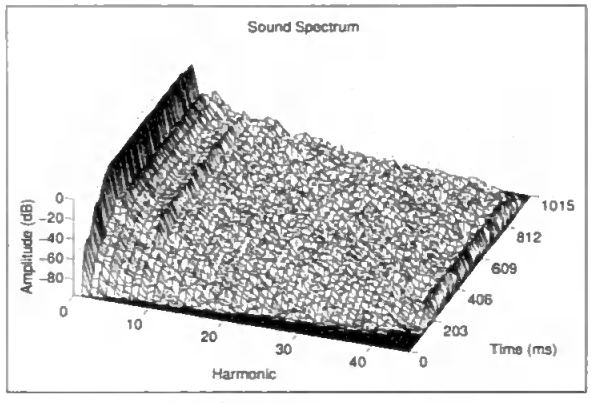


(e) Difference spectrum

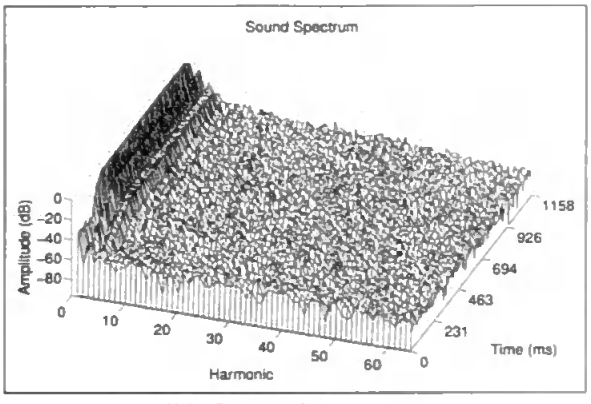


(f) Difference spectrum

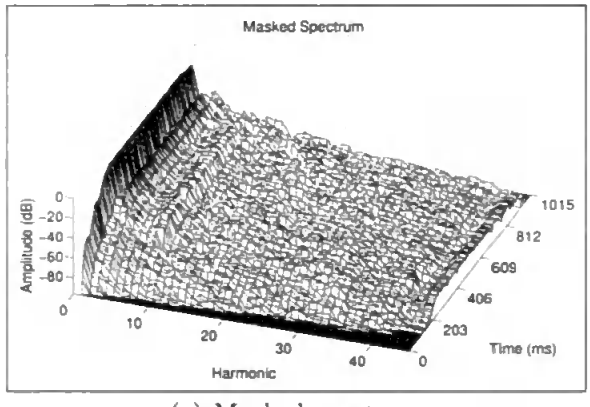
Figure 3.7: Perceptual masking for Basson (left) and Horn (right) sounds



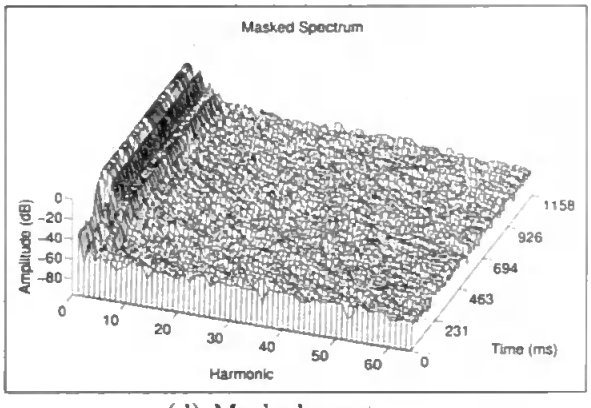
(a) Original spectrum



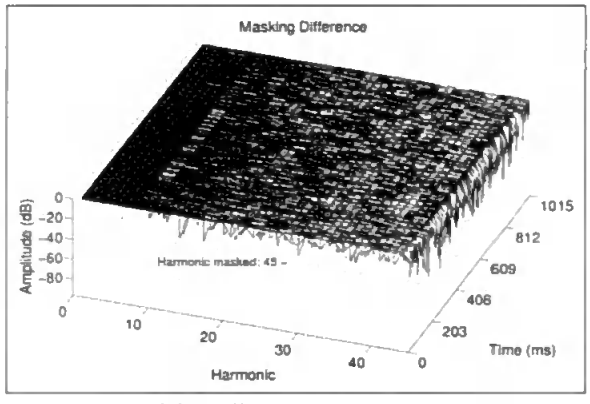
(b) Original spectrum



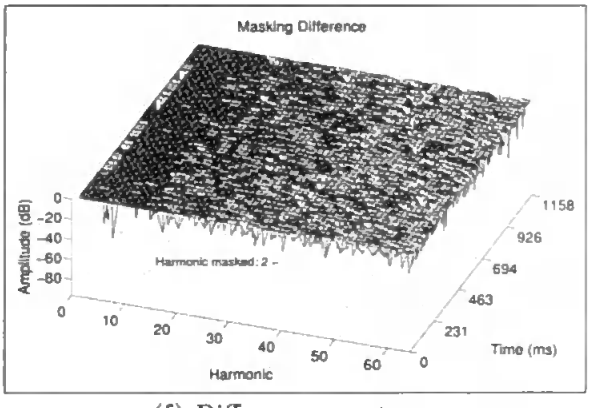
(c) Masked spectrum



(d) Masked spectrum



(e) Difference spectrum



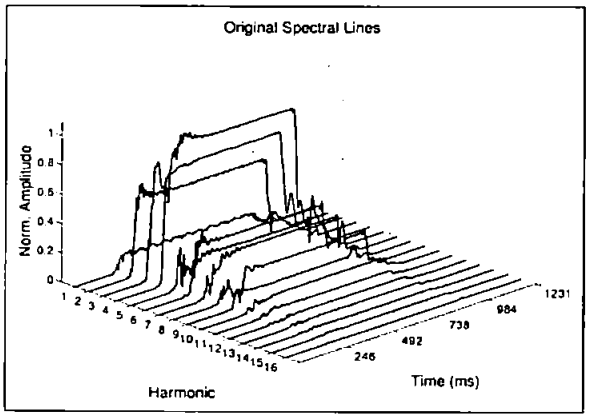
(f) Difference spectrum

Figure 3.8: Perceptual masking for Flute (left) and Principal (right) sounds

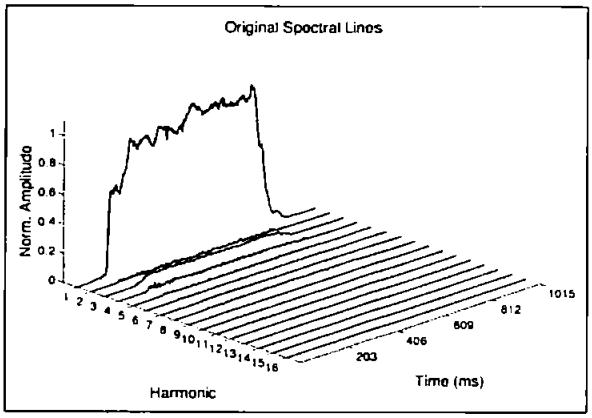
3.3.4 Spectro-Temporal Simplifications

Spectro-temporal simplifications of the harmonic data (reduction both amplitude and frequency) were implemented following the experiments described in [117].

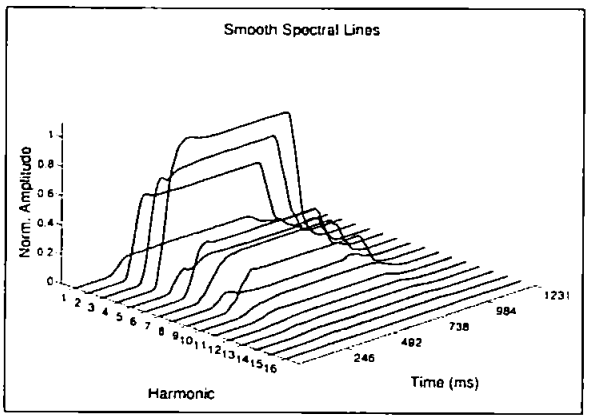
These developments are important because of the two following reasons. Listening tests performed by the audio experts are used to assess the threshold at which these spectro-temporal simplifications become too severe and *perceptible*, and thus *acceptable*. It is often difficult to draw a clear line between the issues of *perceptibility* and *acceptability* in terms of high quality sound synthesis. This often depends on Musicom clients. From a research point of view, the interest lies in the *perceptibility by the audio experts*, i.e. at what level the audio expert considers the audio degradation to be perceptible. From an industry point of view this also includes commercial and practical issues. Spectro-temporal simplifications related to the initial hardware design process of electronic sound generator which has, Musicom ADE included, inherent limitations in its sound synthesis model. These limitations are often related to timing accuracy of the envelopes segments (amplitude and frequency) namely the rate at which the parameters are updated. Smooth and segmented envelopes examples are presented in Figures 3.9 and 3.10 show examples of original harmonic envelopes with smoothed and segmented versions for the Basson, Flute, Horn and Principal sounds.



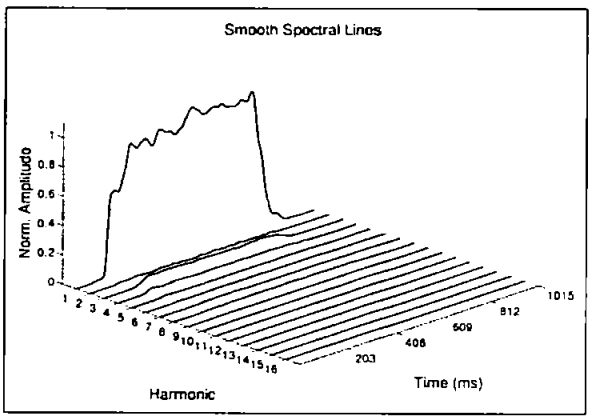
(a) Basson - Original



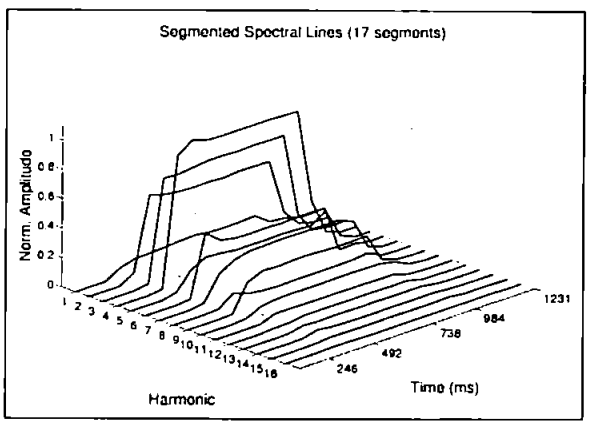
(b) Flute - Original



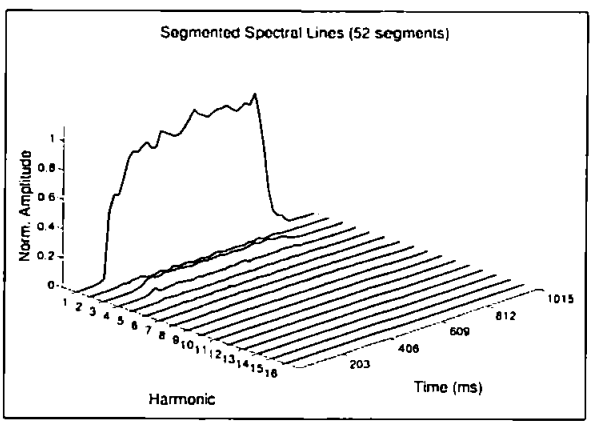
(c) Basson - Smooth envelopes



(d) Flute - Smooth envelopes

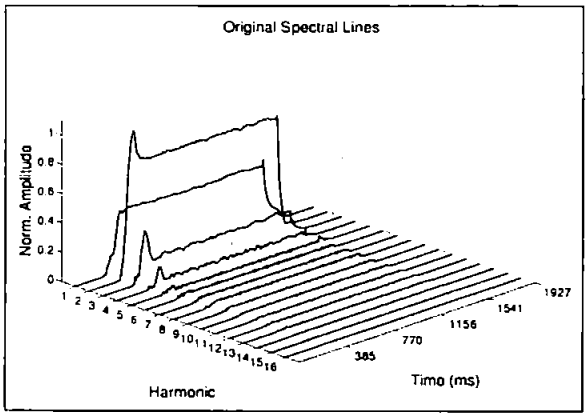


(e) Basson - Segment envelopes

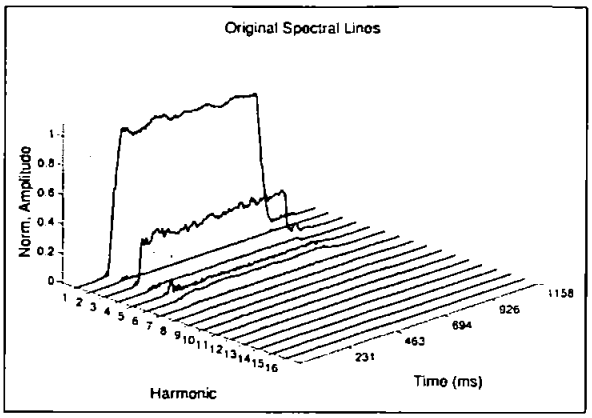


(f) Flute - Segment envelopes

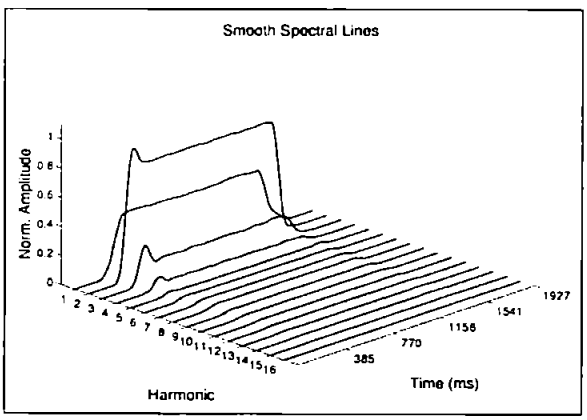
Figure 3.9: Spectro-temporal simplifications on Basson and Flute sounds



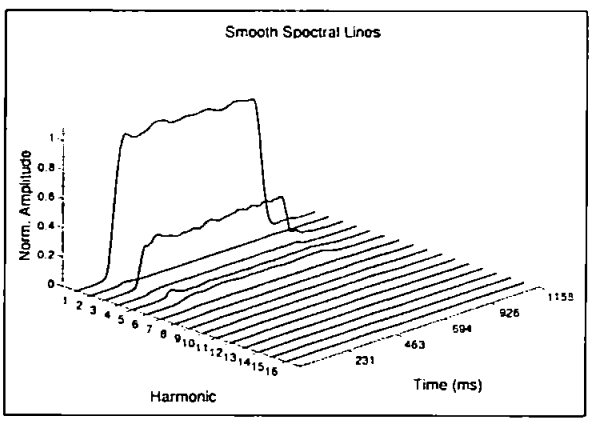
(a) Horn - Original



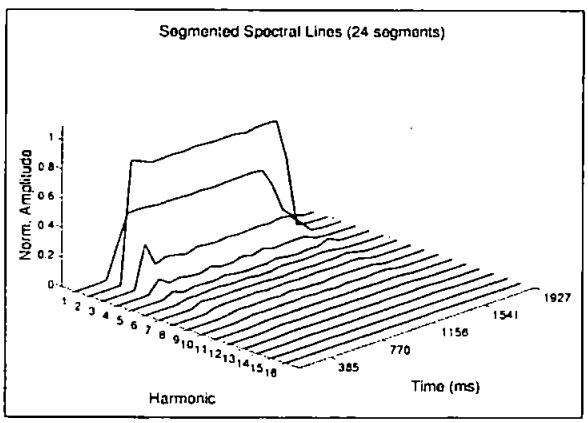
(b) Principal - Original



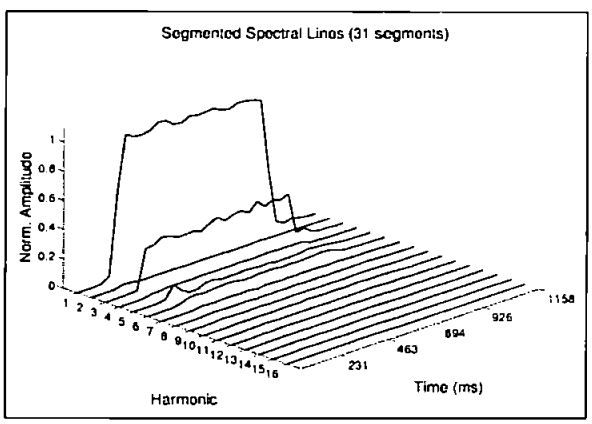
(c) Horn - Smooth envelopes



(d) Principal - Smooth envelopes



(e) Horn - Segment envelopes



(f) Principal - Segment envelopes

Figure 3.10: Spectro-temporal simplifications on Horn and Principal sounds

From these data reduction with spectro-temporal simplifications, we can draw the following conclusions:

- Spectro-temporal simplification can, in some cases, remove most of the original characteristic of the original sound when its harmonic structure is complex and especially in the attack part of the sound (Figure 3.9(a)) often considered as giving the most realism and "life" of the sound [6].
- As an research or engineering problem, data reduction has always been a driver for research, future work should focus on perceptual data reduction with techniques that would perceptually reduce the harmonic data without removing the original sound's characteristic. This could be easily automated with the current system and techniques already described in [61].
- The objective evaluation of audio quality of sounds processed by spectro-temporal simplifications would be an interesting topic to investigate. It would give indications on the trade-off between complexity and sound quality, important for future hardware design of electronic musical instruments. It would also provide a good benchmark technique to compare techniques for algorithmic sound synthesis (including the Structured Audio Orchestra Language (SOAL) used to describe an algorithm producing sounds in MPEG-4 Structured Audio [79]).

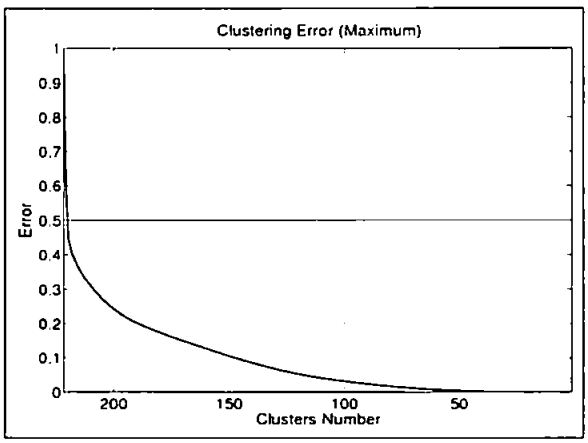
3.3.5 Agglomerative Hierarchical Clustering

Original clustering algorithm used by Musicom was based on a basic Agglomerative Hierarchical Clustering (AHC). The procedure of this clustering algorithm is as follows. First, each harmonic amplitude envelope is assigned to a separate cluster. Then the all pair-wise distances between clusters are evaluated, with a distance matrix built and storing these distance values. The pair of clusters with the shortest distance is found from the matrix and is removed by a process of merging. All the distances from this new cluster to all

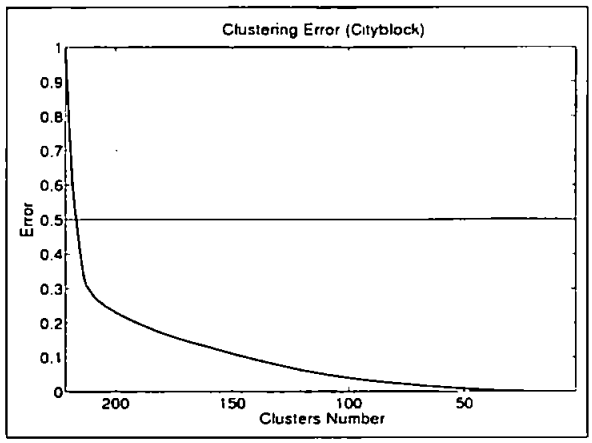
other clusters are evaluated, and the matrix updated. This is repeated until the distance matrix is reduced to a single element. This algorithm provides an easy way to reduce the clustering process to a simple iterative software routine. The two major advantages of using AHC are that 1) it can produce an ordering of the harmonic, which may be informative for data display (See error and dendrogram Figures) and 2) smaller clusters are generated, which may be helpful for discovery. However, the major disadvantages of AHC are that 1) there is no provision for a relocation of harmonics that may have been "incorrectly" grouped at an early stage, thus the final result should be examined closely to ensure it makes sense and that 2) the use of different distance metrics for measuring distances between clusters may generate different final results, in other words, performing multiple experiments and comparing the results is often recommended to support the veracity of the results, in the present case the audio quality of the sound re-synthesised using these clustered harmonics.

The original WinPVoc AHC algorithm was using Euclidian distance and the motivation was to add other type of distance metric methods and assess their effectiveness with the AHC algorithm to finally formulate more appropriate clustering algorithms for harmonic clustering. The mathematical definition of the distance metrics are presented in Table 3.2. In WinPvoc, the distance metric method can be selected from the Preferences DialogBox as shown in Figure 3.16. To facilitate the clustering algorithm development and better understand the issues of clustering all the operations of the AHC algorithm are traced into a log-file. Specifically of interest are which harmonics are selected and merged and the distance metric values calculated deciding this merge, and this for each iteration.

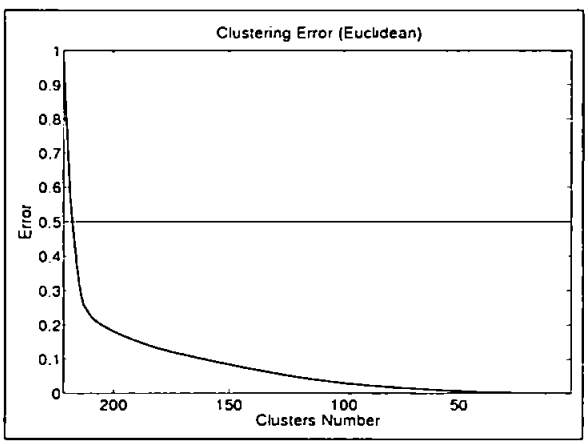
Figure 3.11 shows a comparison of the clustering error introduced by different distance methods used for an AHC (Basson 8' sound example). The error shown is calculated as an accumulative error and it the classic way in clustering to present the algorithm performance. Results of the different distance metric of the AHC are presented in Chapter 6. It



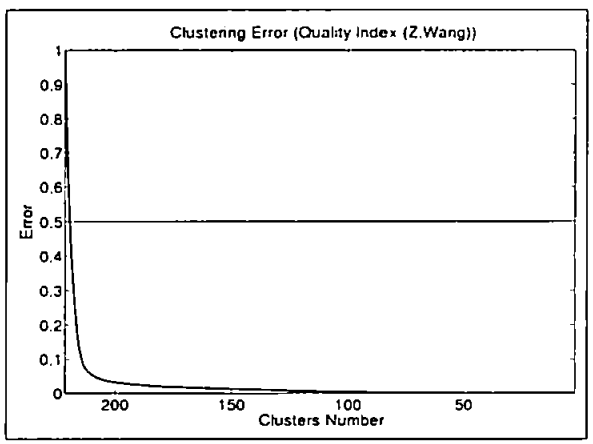
(a) Maximum



(b) Cityblock



(c) Euclidean



(d) UQI [164]

Figure 3.11: Clustering error for different distance metrics

is worth noticing that none of these distance metric are based on human hearing system (i.e. perceptual) and that there is still a need to develop a suitable "perceptual harmonic similarity index" for sound design.

The following discuss each distance metrics of Table 3.2. The Minkowsky distance is one of the most commonly used distance metric in data analysis and statistics. One particular case is the Euclidean distance $\beta = 2$, i.e. taking the square root of the summation error (the error being the difference of the two vectors), originally used in WinPVoc. The Chebychev distance may be appropriate if the difference between points is reflected more by differences in individual dimensions rather than all the dimensions considered

Minkowski distance (Note: Euclidean is when $\beta = 2$)

$$\text{Minkowski}(x, y) = \left[\sum_{k=1}^{N} |x_k - y_k|^{\beta}\right]^{\frac{1}{\beta}} \tag{3.2}$$

Cityblock or Manhattan distance (L1 metric)

$$\text{Blk}(x, y) = \sum_{k=1}^{N} |x_k - y_k| \tag{3.3}$$

Chebychev distance or Maximum metric (L_∞ metric)

$$\text{Max}(x, y) = max|x_k - y_k| \tag{3.4}$$

Correlation distance

$$\text{Corr}(x, y) = \frac{\sum_{k=1}^{N}(x_k - \overline{x})(y_k - \overline{y})}{\left[\sum_{k=1}^{N}(x_k - \overline{x})^2 \sum_{k=1}^{N}(y_k - \overline{y})^2\right]^{1/2}} \tag{3.5}$$

Universal Quality Index

$$\text{UQI}(x, y) = \frac{4\sigma_{xy} + \overline{x}\,\overline{y}}{(\sigma_x^2 + \sigma_y^2)[(\overline{x})^2 + (\overline{y})^2]}, \tag{3.6}$$

where

$$\overline{x} = \frac{1}{N}\sum_{k=1}^{N} x_k, \qquad \overline{y} = \frac{1}{N}\sum_{k=1}^{N} y_k,$$

$$\sigma_x^2 = \frac{1}{N-1}\sum_{k=1}^{N}(x_k - \overline{x})^2, \qquad \sigma_y^2 = \frac{1}{N-1}\sum_{k=1}^{N}(y_k - \overline{y})^2,$$

$$\sigma_{xy} = \frac{1}{N-1}\sum_{k=1}^{N}(x_k - \overline{x})(y_k - \overline{y})$$

Table 3.2: Distance metrics for clustering algorithm

together and it is important to notice that this distance measurement is very sensitive to outlying. As some harmonics can be very noisy, clustering using Chebychev distance can be carried out with smoothed version of the amplitude envelopes of the harmonics (as shown in Figure 3.10(c) or Figure 3.10(d)).

The use of Universal Quality Index distance was inspired from [164] in which a UQI is proposed and evaluated for assessing the quality of JPEG compressed images. Despite its relative simplicity (Equation 3.6), UQI provides better results than the commonly used Mean Squared Error (MSE) and from a signal processing point of view, it is very efficient. UQI is based on the three main factors: the loss of correlation, the luminance distortion and the contrast distortion, which related to images but these features can be easily mapped into audio terms. Recently, Wang and Bovik proposed an improved quality index called Structured SIMilarity (SSIM) Index [165] that extended the research on UQI. Results using UQI for sound quality assessment are presented in the Chapter 6. Further investigations are being carried out as part of investigations into UQI-based sound quality index.

Results of clustering algorithms are often shown using a dendrogram. It consists of many upside-down, U-shaped lines connecting objects in a hierarchical tree. The height of each U represents the distance between the two objects being connected. In the example shown in Figure 3.12, the AHC algorithm is used to model a Dulciana sound. The distance method used is Euclidean and merging process uses the average method. The vertical axis shows the selected cluster of harmonics and vertical axis shows the inter-cluster distance. The dendrogram clearly shows that the sound can be modelled using a combination of cluster 2, with cluster 1 and 3, cluster 4 and 6, and the summation of the other clusters.

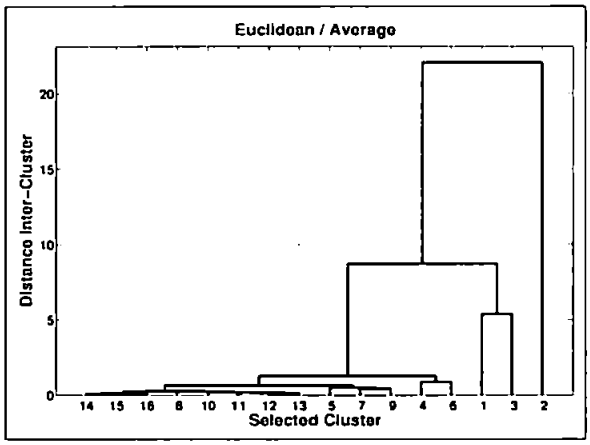


Figure 3.12: Dendrogram - Result of AHC on a Dulciana sound

3.3.6 Intelligent Audio System

The intelligent audio system is the main contribution of this thesis. It is based on a fuzzy model, implemented in software as a fuzzy expert system. It is a rule-based system that processes the audio features, extracted from the sound analysis engine, following rules defined by the audio experts, to generate optimal (sub-optimal) sound synthesis parameters for the sound synthesis engine. Chapter 5 describes in details its development.

A brief description of the system is given here and Fuzzy Logic principles are given in Section 5.3. A conceptual diagram of the Intelligent Audio System is shown in Figure 3.13. The fuzzification process transforms the audio features into fuzzy variables. Fuzzy variables are made of two or more fuzzy sets. Each fuzzy set defines the *degree of membership* (Vertical axis) that a fuzzy variable belong to along the *universe of discourse* (Horizontal axis). A fuzzy inference engine, the rule-based decision making process, applies the fuzzy rules. In this research project audio experts have defined rules that best describe the complex process of pipe organ sound design. These rules are in essence their "audio expertise", based on years of experience working on digital pipe organ sound design. Finally, the fuzzy model output variables are going through a process of defuzzification to generate sub-optimal synthesis parameters used to configure and control the sound synthesis engine. This novel concept was published in [60] for the

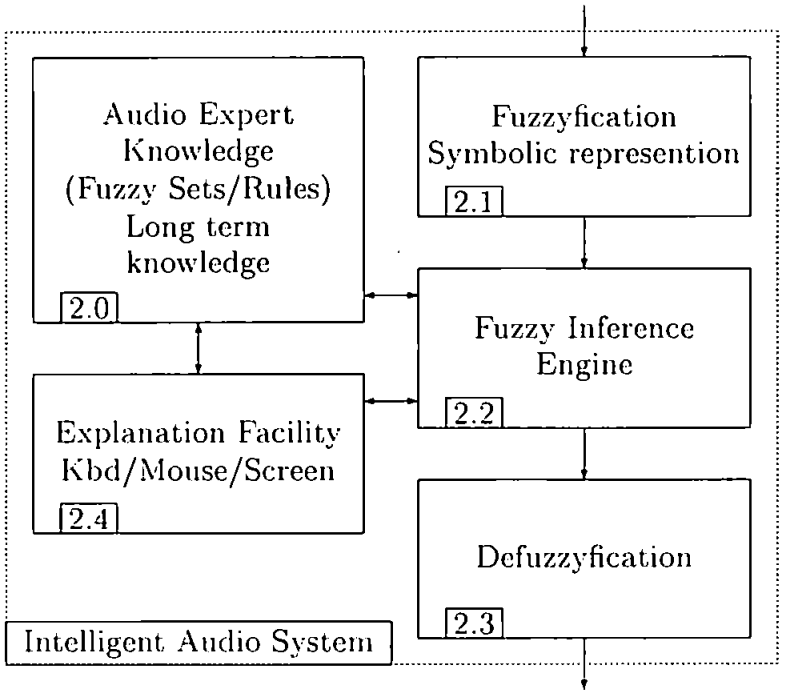


Figure 3.13: Diagram of the Intelligent Audio System

first time in audio and computer music research.

3.4 Sound Synthesis Engine

The Sound Synthesis Engine (see block 3 of Figure 3.1) is the third part of the intelligent audio system. A diagram of the sound synthesis engine is shown in Figure 3.14. It is based on two modules, a software-based module and a hardware-based module. The software-based module is an implementation of the synthesis of the Phase Vocoder which generates sound file in WAVE format. The audio experts can listen to the synthetic sound using the computer running WinPVoc or from the Musicom ADE connected to a digital audio mixer (DR128 from Allen Heath Ltd [4], (Penryn, Cornwall)) with speaker monitors and good quality headphones (Sennheiser HD590). Listening tests allow the audio experts to assess the synthetic sound quality, giving a true performance index of

the intelligent audio system.

3.4.1 Software Engine

The sound synthesis engine implements the re-synthesis part of the Phase Vocoder using overlapping windowed IFFT [129]. The system is a *true* analysis/synthesis modeling system. The window size is the main parameter adjustable of the sound synthesis engine, allowing to find trade-off between time and frequency accuracy of the Phase Vocoder algorithm [130]. The other parameter relates to the phase that set to be ignored (the phase information in the synthesis was always used in our experiments). Very little has been published on the perceptual effects of the phase in sound modeling else than in [5] and recently in [41]. Recent work by Röbel on transient detection and preservation [135][136] will be used to improve the accuracy of both sound analysis and synthesis engine, and thus the intelligent audio system.

3.4.2 Musicom ADE System

The sound synthesis engine can also be used to configure the Musicom ADE[‡] (hardware tone generator). With this option, it is possible to test the results of the research directly and in real-time. The Musicom ADE is the acronym for Active Digital Environment, a digital sound synthesis environment with a real-time interactive engine for producing music (electronic pipe organs, piano, bells and even Hammond organ [158]). Figure 3.18 shows a picture of the Musicom ADE used in this research.

The WinPVoc software can generate specifications for the Musicom ADE system. These are a set of data files describing the physical (hardware), sound characteristics of a pipe organ system, and the way the resources are allocated. These hardware resources consist of generators for noise, envelopes for amplitude and frequency, as well as spectral

[‡]ADE is a registered trademark of Musicom Ltd

shapes for creating wavetables, mapping tables use for the control of the sound and overall instrument. The real-time allocation of the resources are defined and loaded by software, allow full real-time control of the sound's parameters and whole pipe organ. More information can be found in the VASM user manual [99].

Traditionally, the audio experts would have to write the scripts by hand in the Voicing Assembly (VASM) language [96]. Nowadays, WinPvoc and PipeSpecBuilder Musicom software tools can create automatically all the necessary specification files (thus sparing users the reading and understanding of the complex VASM syntax). Recent updates of the VASM language specifications [99] follow the recent associated software and hardware developments of the Musicom ADE recent versions .

Historically, the VASM language was originally developed at Bradford University from pioneer research of Peter Comerford to formally specifying electronic organ systems [22]. Comerford has been developing the Bradford Musical Instrument Simulator (BMIS) for many years, looking at simulating pipe organs with additive synthesis [23][24]. The original design of the Musicom ADE was based on Comerford original patent [21], but has been continuously (and independently) improved over the last 15 years to the current version (ADE4.5) which is today celebrated by many organists all over the world.

3.5 Sound Quality Assessment Engine

The intelligent audio system uses an original sound to produce a synthetic sound. The performance of the system is assessed by comparing the audio quality of the synthetic sounds relative to the the original one. This is done 1) subjectively with listening tests performed by the audio experts and 2) objectively using a novel method with the ITU-R BS.1387 PEAQ algorithm [81]. Results are presented in Chapter 6.

With the first method, the final sound quality assessment was carried out by the audio experts at Musicom Ltd facilities (Bideford, Devon). The author, even with more than

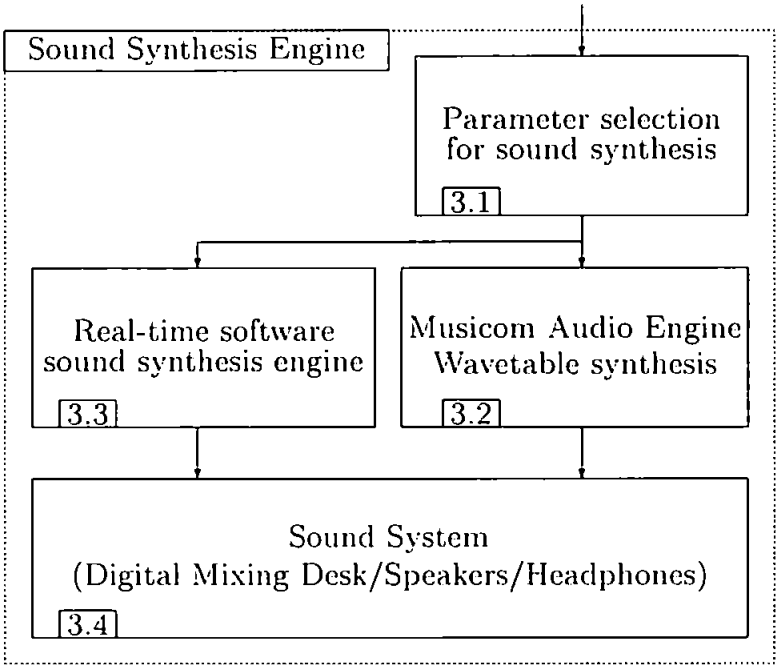


Figure 3.14: Diagram of the Sound Synthesis Engine

5 years experience in audio research, does not have such "Golden ears"[†] [147][146] and experience required for this task.

To facilitate the collection and management of Subjective Difference Grade (SDG) scores, a Subjective Difference Grade (SDG) DialogBox has been added to WinPvoc (As shown in Figure 3.17). The listening test results can be sent to the WinPVoc developer via email containing the audio expert SDG score (using the scale shown in Table 2.1 from ITU-R Rec. BS.1116 [80]), together with some comments and the WAVE files as attachments. This feature is used extensively and provides an easy solution to SDG score database collection. This will be valuable for future work as described in Section 7.5.

The second method uses the PEAQ algorithm and provides an Objective Difference Grade (ODG) score based on the scale shown in Table 2.2. The motivation for using such method was that subjective listening tests are known to be expensive, time consuming,

[†]People who possess extremely acute hearing

requiring specialised sound facilities and should, in the context of the ITU framework, require a large number of subjects to obtain the required accuracy [80], even if the two audio experts were, most probably the two best individuals for this task, as they have extensive experience in subjective quality assessment or pipe organ sounds. Thus the Opera system[†][91] (pictured in Figure 2.8), provided the ideal tool to implement the novel approach and automate the sound quality assessment engine as an objective audio quality metric of the sound quality based on models of human perception [180][155].

The audio experts have welcomed both sound quality assessment methods with a preference for the first one. The suitability of using PEAQ in the context of this research project is discussed in Section 7.4. Recently, Benjamin [11] described an evaluation of digital audio artifacts using PEAQ, measuring the audio degradation caused by analog to digital converters, digital to analog converters, and sample rate conversion units.

3.6 Optimisation Loop

The intelligent audio system has, like an human expert always learning from trial-and-error, a feedback loop used for optimisation. Currently, this optimisation loop only controls the audio feature processing part (block 3 in Figure 3.1, however future work will extend this to both sound analysis engine and sound synthesis engine parts. The sound analysis engine could have different types of analysis techniques and depending on the results achieved by the intelligent system, more accurate analysis techniques could be selected. The same concept would apply to the sound synthesis engine, using alternative sound synthesis techniques described in Section 2.4.

From a research and engineering point of view, an optimal solution always represents the final goal, and this optimisation loop is very natural in that sense. This optimisation can also be seen similar to the way humans adapt, the audio experts always seeking the

[†]Opticom GmbH, Germany

highest sound quality. The learning process by *trial-and-error* generates new knowledge that can be taken into account, refined over time, and used to avoid being trapped in the same situation (i.e. famous quote of "*learning from mistakes*").

This feature, most importantly, makes the system *self-tuning* which is very important as an automated research context. This research project is the first attempt in computer music to develop an intelligent audio system that can generate sounds of the highest audio quality like the audio experts can. This optimisation loop provides a way to automate to this process.

3.7 Research Tool WinPVoc

The intelligent audio system has been implemented and integrated into WinPVoc (short for original Windows Phase Vocoder) originally developed at Musicom for in-house sound analysis/synthesis and sound design. A screen shot of WinPVoc is shown in Figure 3.19. From its original version, WinPVoc has been extended in many ways [58]. More specific to this research project, the sound analysis engine generates MATLAB † scripts and binary data from the Phase Vocoder analysis. Time-varying evolution of the harmonics in both amplitude and frequency (with important fundamental frequency) are stored in the binary files. MATLAB scripts implement all the various functions discussed in this chapter, including the extractions of audio features (described in more detail in Chapter 4. These MATLAB scripts make use of Digital Signal Processing (DSP) [78], statistical analysis and Fuzzy Logic [139], all functions are available from MATLAB toolboxes. Additionally, the MATLAB environment provides graphic capabilities to plot in 2D and 3D data into figures that can be exported as images (JPEG) or Encapsulated Post-Script (EPS) files and included into LaTeX† format documents for example. Combined with LaTeX, the MATLAB environment is probably the most powerful research tool available to researchers, and

†Matlab is a registered trademark of The MathWorks, Inc.
†LaTeX- http://www.latex-project.org

WinPVoc benefits greatly from it. Updates on WinPVoc (current version is 0.483) are usually emailed to the audio experts on a fortnight basis. More information on WinPVoc can be found in [58].

3.8 Summary

In this chapter, an intelligent audio system for the sound design using was presented. The system provides an innovative AI-based approach to sound synthesis parameter optimisation. The system developed in collaboration with two audio experts is used to analyze acoustic recordings of musical instruments, extract salient audio features and process them using rules that have been defined by the audio experts, in order to generate parameters for a sound synthesis engine. The whole process is done in a manner that attempts to mimics human audio experts. To the author's knowledge, this is the first attempt in computer music to capture and exploit, explicitly, knowledge from audio experts for sound design and modeling musical instruments.

Developments have been described on manual clustering, ranking and permutation, perceptual masking, spectro-temporal simplifications and algorithm clustering all carried out to gain, initially, better understanding of the sound analysis, synthesis and quality assessment, and help preparing the design and development of the *fuzzy model of audio expertise*. This fuzzy model of audio, the main contribution of the research work, is presented in Chapter 5.

The concept of the intelligent audio system, together with preliminary results, was first published in [60] and presented at the Audio Engineering Society convention in New York in December 2001. More results on the "Objective Prediction of Sound Synthesis Quality" were published in October 2003 in [61], in which the focus was on the audio quality assessment engine of the intelligent audio system and its use for automated perceptual assessment of sound synthesis. Appendix C.2 and Appendix C.3 provide copies of the

two AES Preprints.

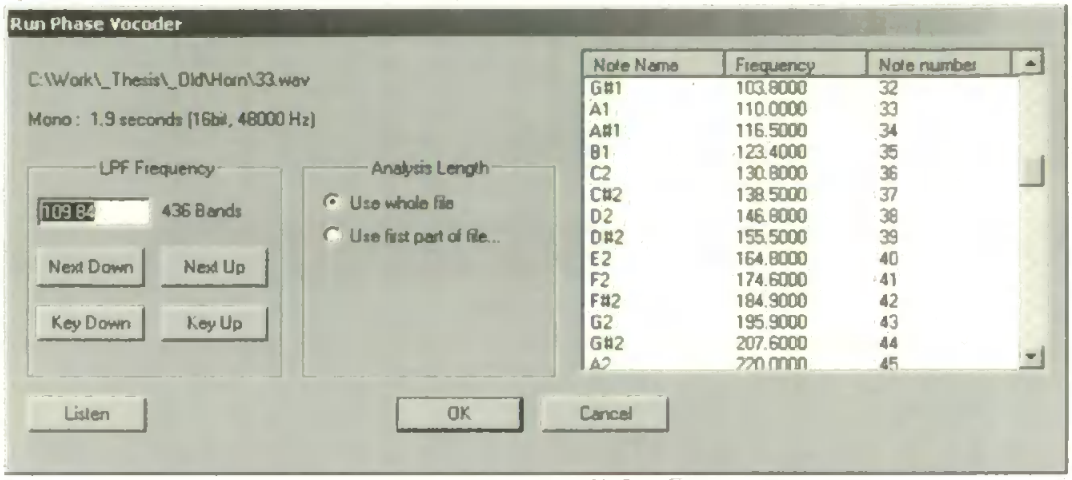


Figure 3.15: WinPVoc Sound Analysis DialogBox

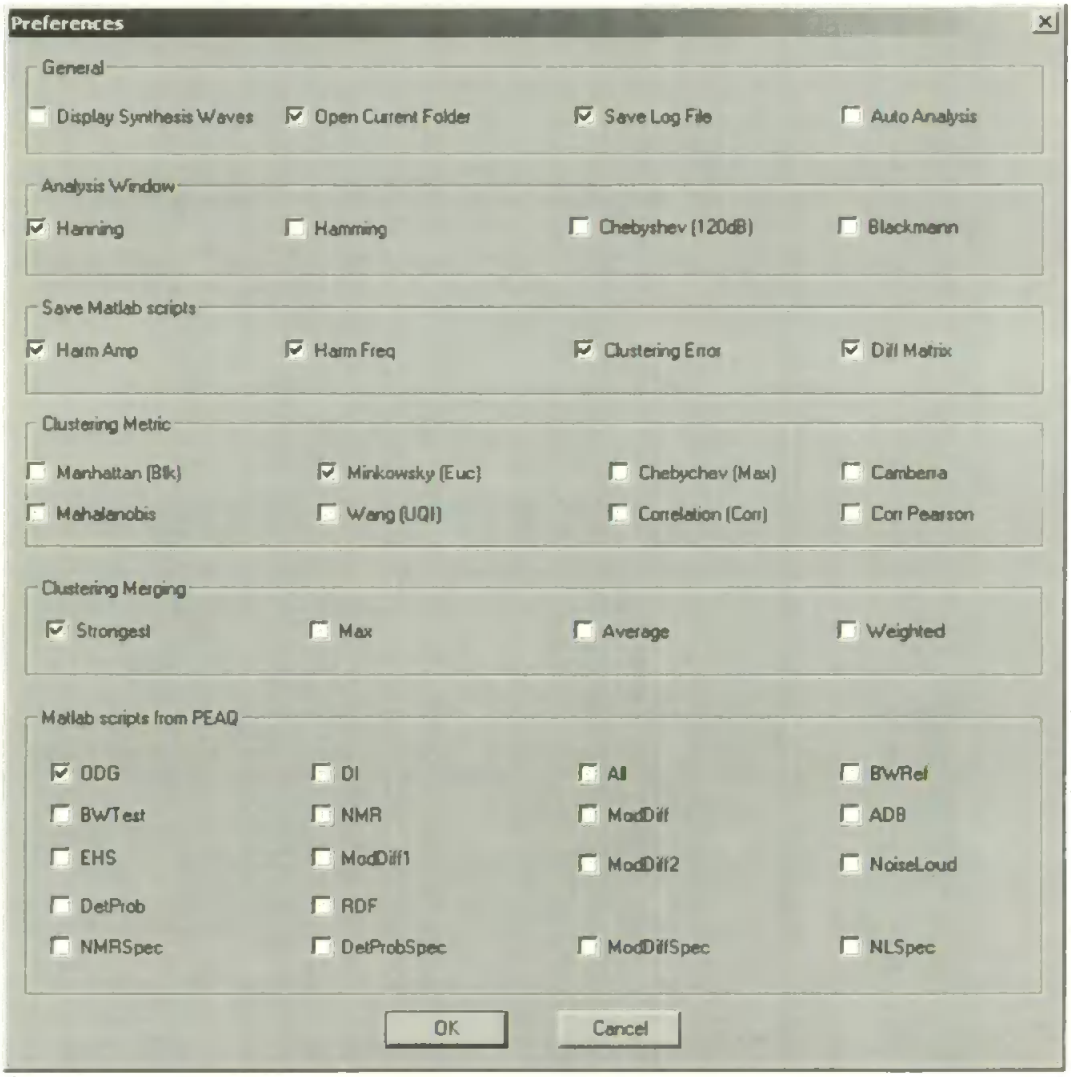


Figure 3.16: WinPVoc Preferences DialogBox

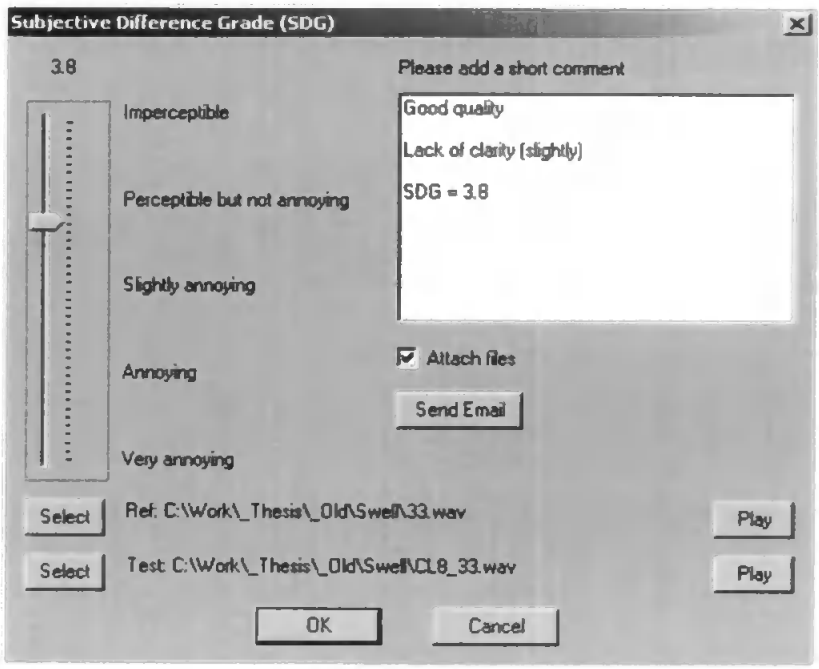


Figure 3.17: WinPVoc SDG DialogBox

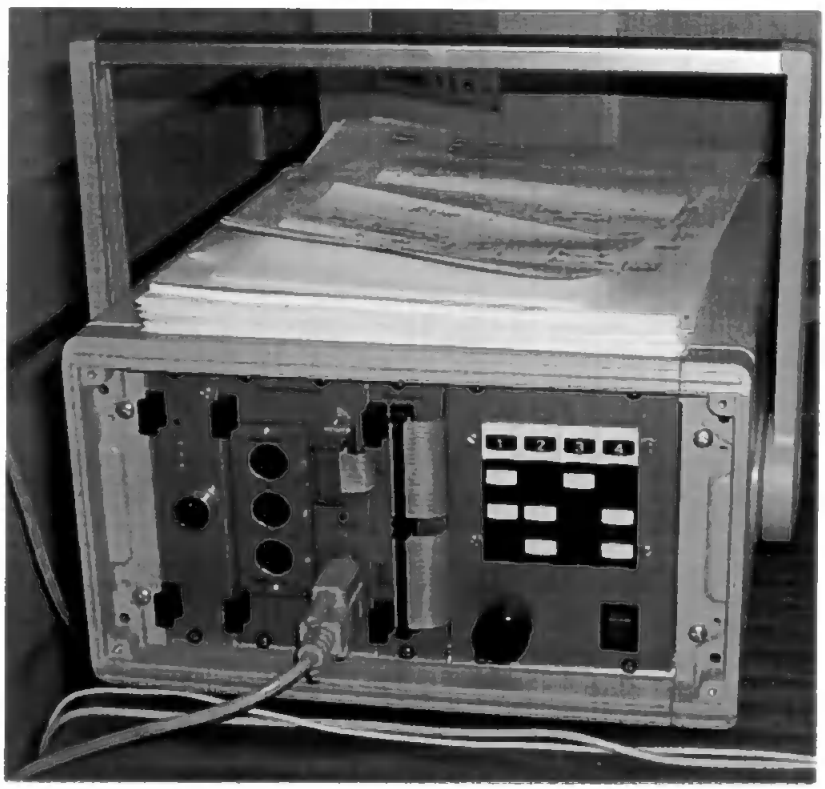


Figure 3.18: Musicom ADE System

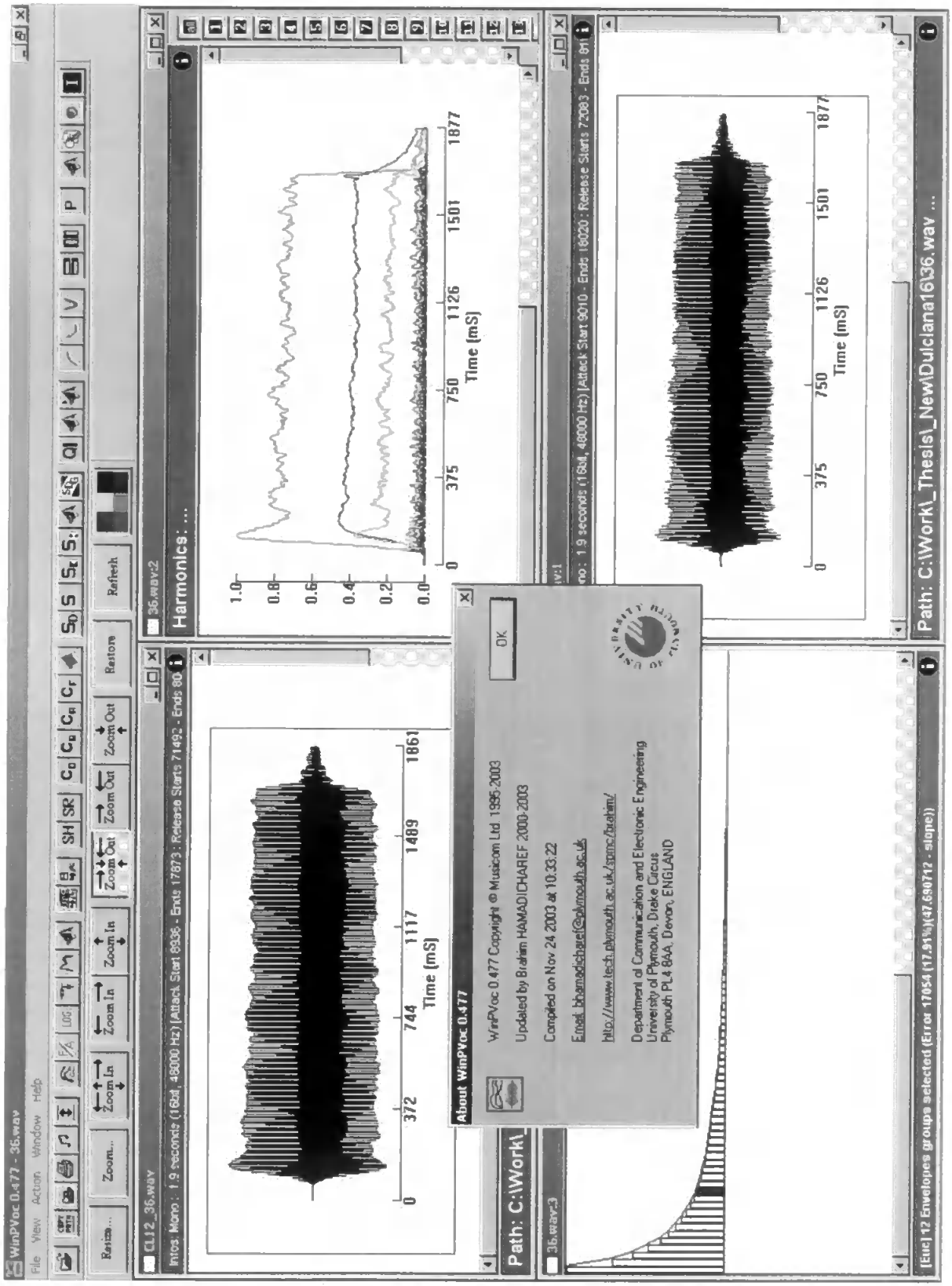


Figure 3.19: Research Tool WinPVoc

Chapter 4

Sound Analysis and Audio Features Extraction

4.1 Introduction

This chapter is concerned with the sound analysis engine, front part of the intelligent audio system and more particularly with the audio features and their extraction. Each audio feature is described with the method used to extract and illustrate examples. The implementation was made as MATLAB scripts. These audio features are used as inputs of the fuzzy model developed with the audio experts and topic of the Chapter 5.

4.2 Pre-processing

As stated previously, certain sounds required some adequate pre-processing. First, stereo wavefile have to be converted into mono wave files as the current system does not deal with sounds in stereo or multi channels format. This conversion can be done in two ways, either by selecting which channel to work on (left or right) or by mixing both channels as a weighted sum (adjustable weights in DialogBox). The sound analysis engine then normalizes the sound to a maximum level (full scale of 16-bit PCM).

With difficult recording conditions in large acoustic places like church and cathedrals, pipe organ sounds can suffer from background noise and thus noise reduction can be

required. One important issue with background noise in the context of this work, is that pipe organ sounds are very breathy and noise reduction tools have difficulties to accurately distinct what is background noise from what is part of the sound itself. The first few milliseconds of the sound can often help to obtain a characteristic "noiseprint" (i.e. spectral signature of the background noise). When required noise reduction has been left to the audio experts to deal with as they have the right tools (Sound Forge Studio 6.0 with Noise Reduction DirectX Plug-In 2.0) and extensive experience in the art of recordings pipe organs. The sound database provided both clean and noise-corrupted sounds, which reflect real audio data.

Noise reduction using spectral subtraction algorithms have been implemented in MATLAB with a limited number of experiments carried out, but results are not considered satisfactory and more work is required with a help of the audio experts. Investigations into statistical (Bayesian) model-based approach to noise reduction techniques [48] and audio signal enhancement based on psychoacoustic principles [167] for pipe organ sounds, will be considered in future work to identify best suitable noise reduction method for cleaning pipe organ sounds, and to be applied to a large sound database.

4.3 Audio Features

The sound analysis engine extracts the audio features from the time-varying harmonic components of the Phase Vocoder. These audio features consist of the attack, decay and release time, the amplitude envelope with amplitude modulation, the harmonics distribution, the frequency envelope and the frequency modulation, the pitch, some modified Tristimulus parameters, the brightness and noise factor.

4.3.1 Attack, Decay and Release Time

Rise or attack time can be defined as "the time taken for a signal to rise from silence to full intensity", with the release time, on the other hand, to be "the time taken for a signal to decrease from a sustained amplitude to silence". The end of the attack time usually corresponds to the start of the sustain portion of the sound, or can shown like in the case of Figure 4.1(e) (Horn sound), a additional decay portion. This additional decay is typical of sounds with harmonic overshooting i.e. having a large peak before their sustain part.

The attack segment is very important perceptually and has been recognised to provide the most important feeling of *realism* to the sound perception. In [174] for example, Yuen and Horner used a hybrid combination of sampling and wavetable synthesis to re-create piano tones, with sampling synthesis (see Section 2.4.5) for the attack portion and wavetable for the sustain portion (See also Section 2.4.8).

Using the techniques described in Section 3.2.3, audio features such as the attack, decay and release time are extracted and expressed as a normalised value (1.0 corresponding to the full length of the sound). The sustain portion is considered to be between the end of the attack and start of the release segments. In WinPVoc, these attack and release segments can also be defined manually using the mouse. The audio experts often prefer to over estimate the attack segment. An automation of audio features extraction, like the one presented here, is preferable for work on large sound database.

Figures 4.1(a) to 4.1(f) show illustrations of these attack and release time (with thick vertical markers) on the sustain level (dotted horizontal line). One particular case is the Horn sound example for which the second harmonic presents a large *overshoot* (See Harmonic distribution Section 4.3.3), giving a very *percussive* characteristic to the sound.

The overall attack time is calculated as either the maximal value or mean of the 8 first harmonics attack time (a weighted mean solution make results more robust). For

the release time, it is easier as the majority of the harmonics have similar release times thus a mean value is considered to be the best choice. Harmonics usually fall in a very similar manner to the ones shown in Figure 4.1(f). The Basson example in Figure 4.1(b) could indicate some slight post-resonance effect in the recording, and this is often obvious when listening to the sound.

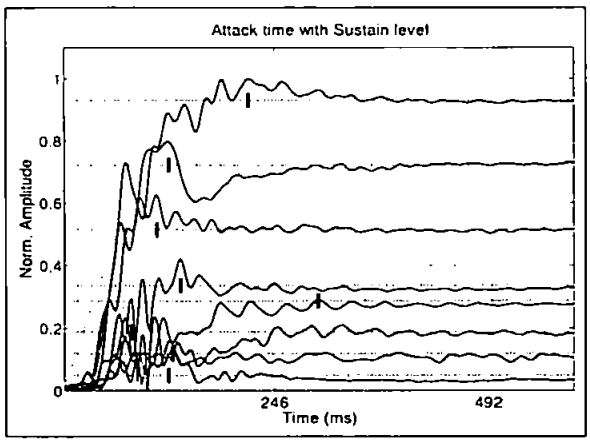
MATLAB script to extract timing information (attack, decay and release) and sustain level is *timing.m*.

4.3.2 Amplitude Envelopes and Modulations

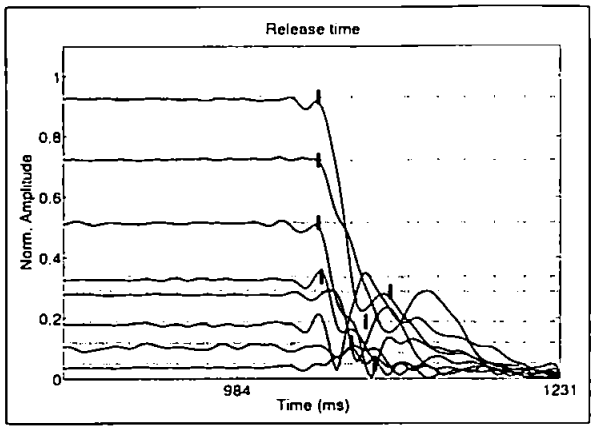
Using the timing information extracted from the previous section, the modulations in amplitude of the harmonics can be assessed. These have been found very important. Figure 4.2(a) and Figure 4.2(b) show the very complex amplitude envelopes and modulations for Basson $8'$ and Metal Trombone $16'$ sounds. Figure 4.2(c) and Figure 4.2(d) shows a simpler case of a Principal sound and the overshoot of few harmonics of the Dulciana sound.

Visually these audio features provide informations to the audio expert in terms of sound characteristic, sound structure, synchronisation between certain harmonics, the stability of harmonics in sustain portions, etc. The amplitude modulations "window of interest" is highlighted automatically (dash lines rectangles), allowing a closer inspection (zoom feature of MATLAB figures).

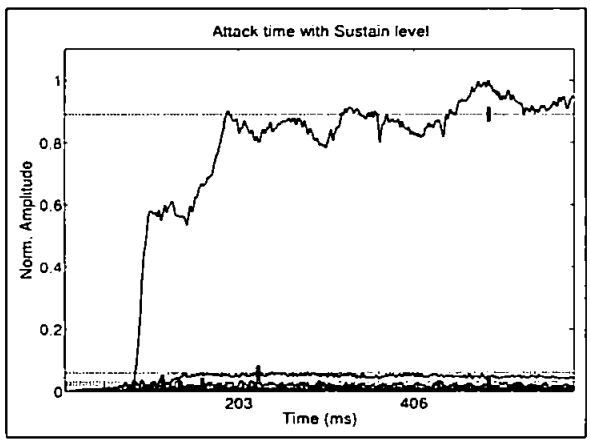
Spectro-temporal simplifications of the harmonic amplitude envelopes, as described in Section 3.3.4, confirmed the importance of these modulations as smoothing them often removes the original sound characteristic. These modulations are what the human ear is sensible to and are what is called in psychoacoustics *Just Noticeable Sound Changes* [180]. Description of the ITU-R BS1387 PEAQ algorithm in Section 2.5.3, also showed amplitude modulations to be used as the 3 first model outputs variables of the perceptual model (model of the human ear) used as inputs to the cognitive model (model of the



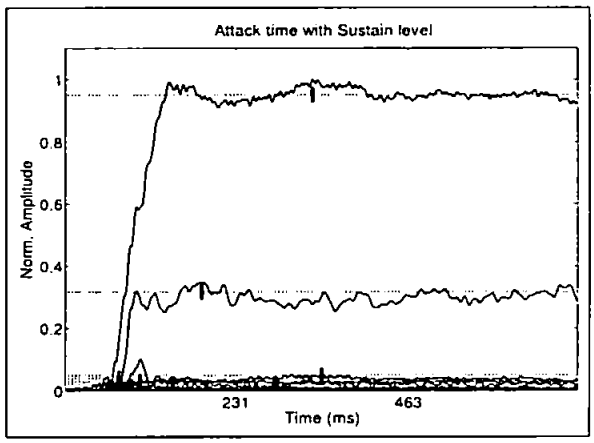
(a) Basson Attack



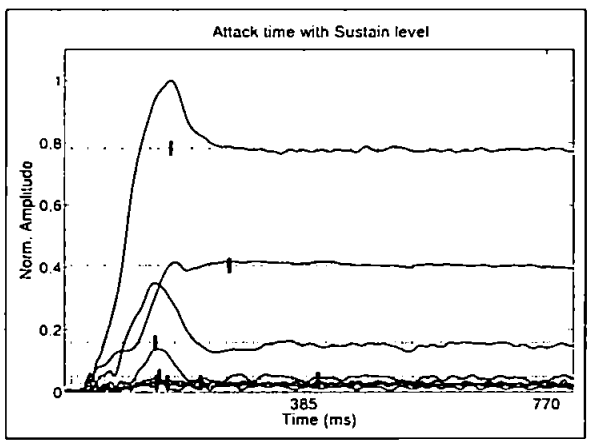
(b) Basson Release



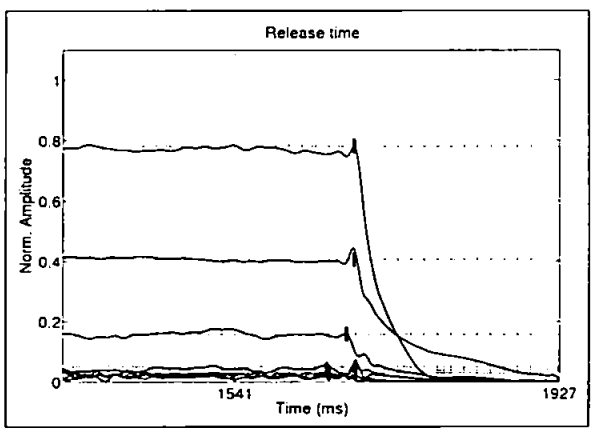
(c) Flute Attack



(d) Principal Attack



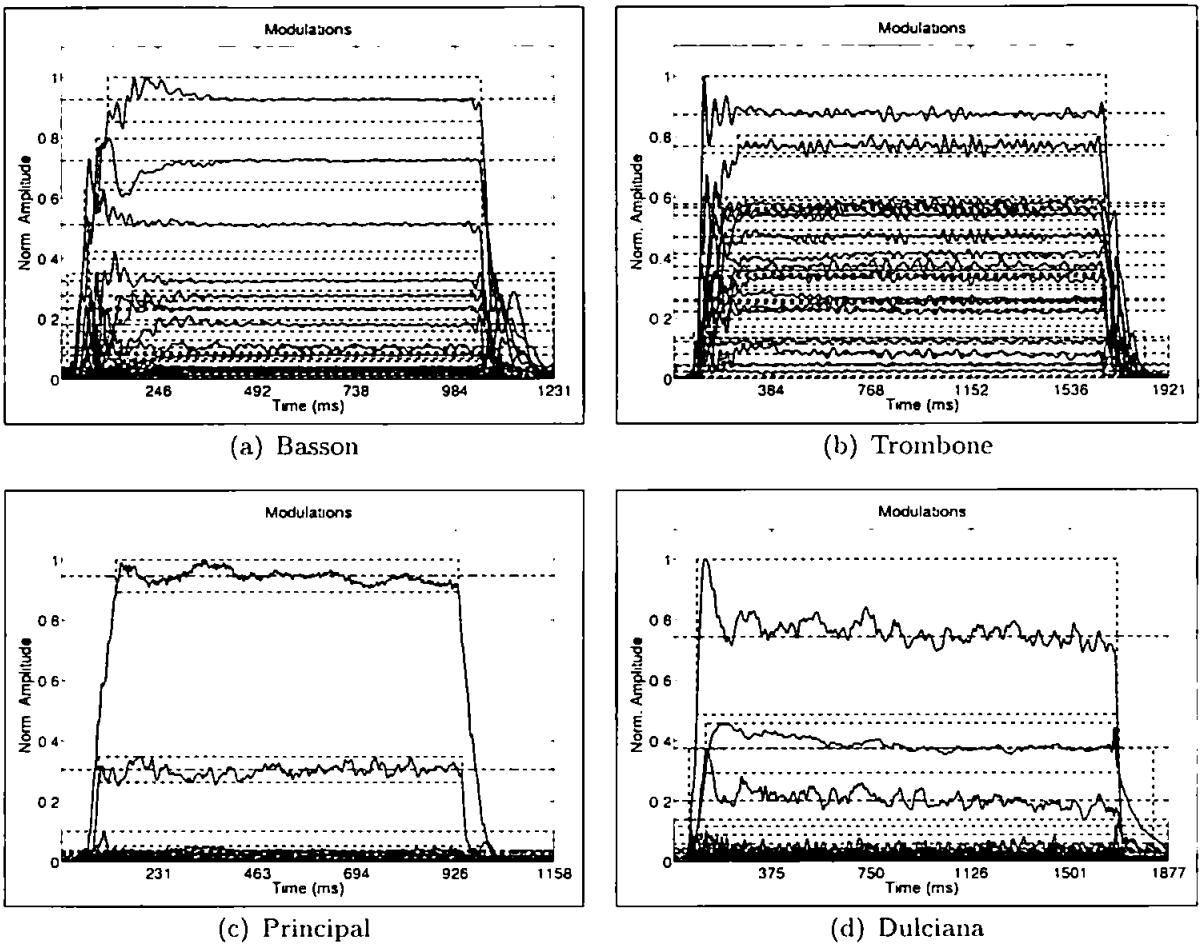
(e) Horn Attack



(f) Horn Release

Figure 4.1: Attack and release of pipe organ sounds

judgement behavior of the test subject). The audio experts emphasised the importance of amplitude modulation to obtain high quality sound synthesis.



(a) Basson (b) Trombone

(c) Principal (d) Dulciana

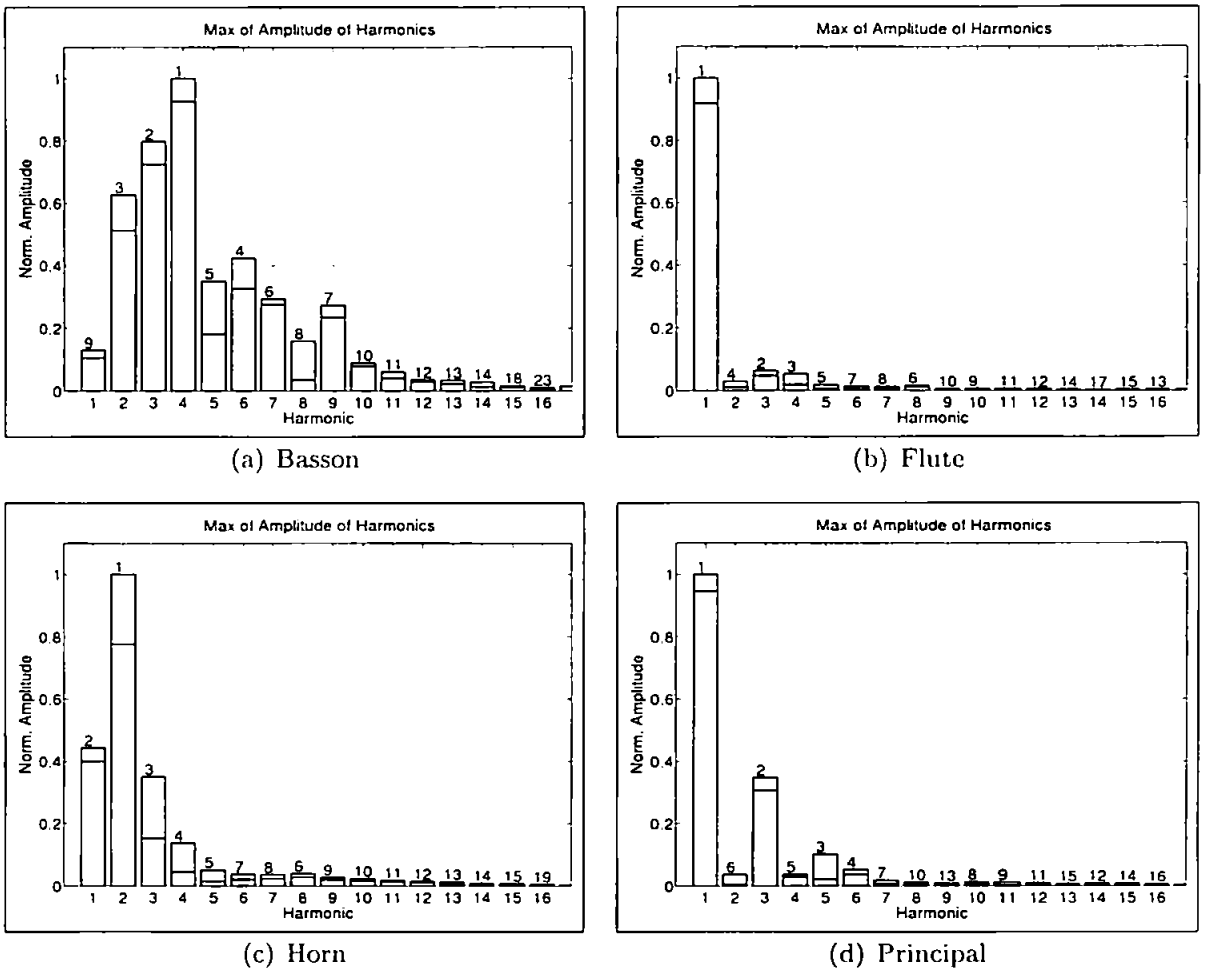
Figure 4.2: Amplitude Envelopes and Modulations

Other time-evolution examples were shown in Figures 3.4, using linear and logarithmic scales in Figures 3.5 (Section 3.2).

Amplitude envelopes and modulations are extracted using The MATLAB script *orgAmp.m*.

4.3.3 Harmonic Distribution

The definition of a harmonic is "a frequency component that occurs as one of a number of such components in a spectrum in which the frequency of every component is an integer multiple of a low (not always the lowest) frequency called the fundamental frequency". A harmonic is, therefore, always a *harmonic of* some particular fundamental frequency. The Phase Vocoder extracts the time-varying evolution of the harmonics (in



(a) Basson (b) Flute

(c) Horn (d) Principal

Figure 4.3: Harmonics Distribution of pipe organ sounds

both amplitude and frequency domains). This section is concerned with the distribution of these harmonics (i.e. relation to each other). Two issues are of particular interest: the maximum value and the sustain level, these are estimated using techniques described in Section 3.2.3).

In all this thesis, the harmonic distribution is referred as the histogram of the harmonic amplitude envelopes as shown in Figures 4.3(a) to 4.3(d) for Basson, Flute, Horn and Principal examples and Figures 4.4(a) to 4.4(c) for Gamba, Oboe, and Dulciana sounds. At first view they are all very distinct and thus can provide the audio experts the following information:

- The first harmonic is not always the highest (as shown in Figure 4.3(a) and Figure 4.3(c)). If the highest harmonic is considered as the fundamental then in these case the lower harmonics can be considered as sub-fundamental (Basson and Horn sounds).

- The overall shape of the distribution gives some indications about the sound complexity as to its spectral structure. A flute sound is very clear, pure and simple (one predominant 1st harmonic) whereas a Basson sound is much richer (distribution centered on the 4^{th}). The Dulciana sound has many harmonics with much modulation (2^{nd}, 3^{rd}, 6^{th} and 7^{th} harmonic).

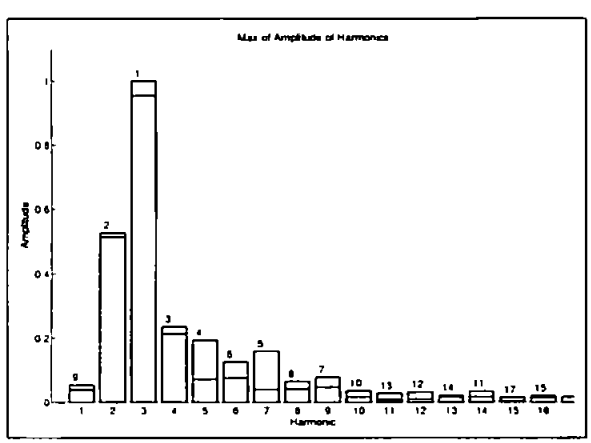
- The ratio between the odd and even harmonics can be also distinctive. One particular example is the Principal sound (Figure 4.3(d)) which has predominantly odd harmonics. The reason being that it was produced using a *stopped pipe*.

- Comparison between the maximum and sustain levels (bar inside the histogram) can indicates "overshoot" of harmonics (from the proportion by which the maximum level is higher than the sustain level). Sounds like the Horn (Figure 4.3(c)) and Dulciana (Figure 4.4(c)) present obvious harmonic overshoots.

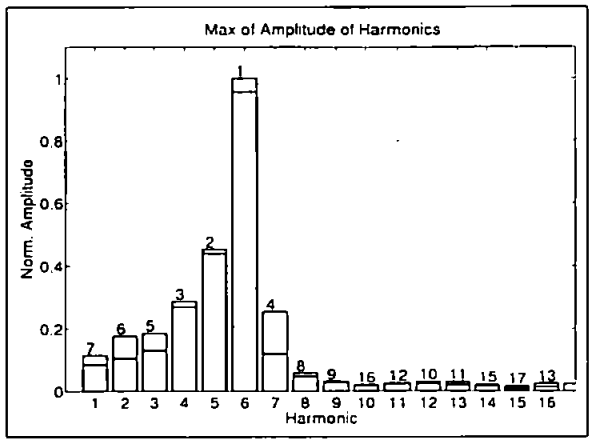
Statistics on the harmonic distribution audio feature on a large sound database, helps in the elaboration of the descriptive report using linguistic terms of the pipe organs from the audio experts (See Section 3.2.4).

Harmonic distributions do share similarities with modified Tristimulus parameters described in Section 4.3.4. However, these harmonic distributions only shown the maximal and sustain level, where as modified Tristimulus parameters are calculated along the time axis.

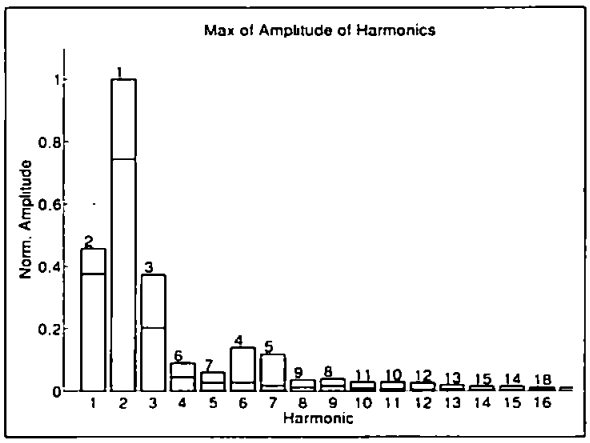
The MATLAB script to harmonic distribution is named *harmBar.m*.



(a) Gamba



(b) Oboe



(c) Dulciana

Figure 4.4: Harmonics Distribution of pipe organ sounds

4.3.4 Modified Tristimulus Parameters

Using the amplitude time-varying envelopes, the sound analysis engine also calculates audio features called modified Tristimulus parameters [102]. These include T_1 the 1^{st} modified Tristimulus, T_2 the 2^{nd} modified Tristimulus, and T_3 the 3^{rd} modified Tristimulus, as well as two additional parameters Odd, the contents of the odd harmonics (without the first one) and Even, the contents of all the even harmonics. Equations 4.1 and Equations 4.2 define the audio features mathematical expressions. These parameters are normalised as $T_1 + T_2 + T_3 = 1$ and $T_1 + Odd^2 + Even^2 = 1$. Illustrative examples are

given in Figures 4.5(a) to 4.5(d).

$$T_1 = \frac{A_1^2}{\sum_{i=1}^{n} A_n^2} \qquad T_2 = \frac{\sum_{n=2}^{4} A_n^2}{\sum_{n=1}^{N} A_n^2} \qquad T_3 = \frac{\sum_{n=5}^{N} A_n^2}{\sum_{n=1}^{N} A_n^2} \tag{4.1}$$

$$Odd = \frac{\sqrt{\sum_{k=2}^{L} A_{2k-1}^2}}{\sqrt{\sum_{i=1}^{N} A_n^2}} \qquad Even = \frac{\sqrt{\sum_{k=1}^{M} A_{2k}^2}}{\sqrt{\sum_{i=1}^{N} A_n^2}} \tag{4.2}$$

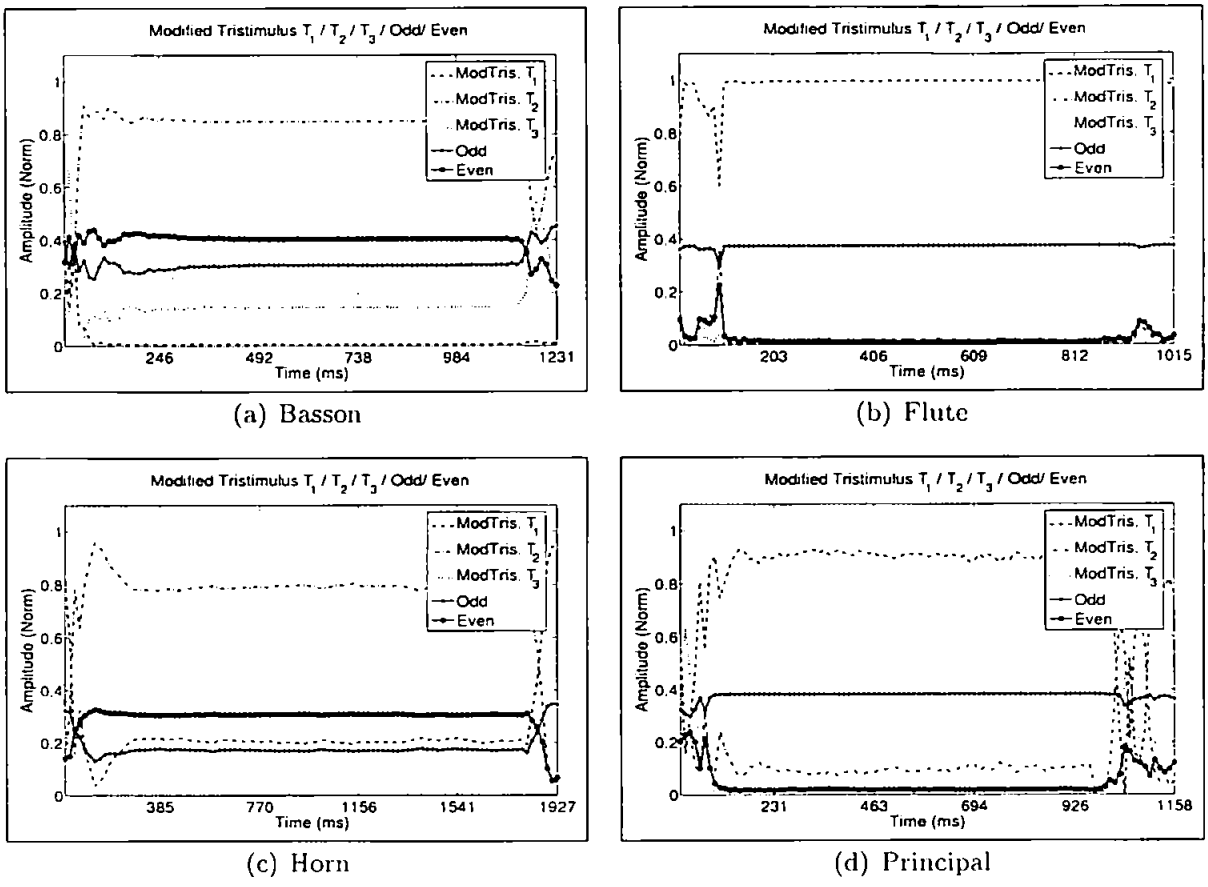
where L = integer value of $\lfloor N/2+1 \rfloor$ and M = integer value of $\lfloor N/2 \rfloor$.

These audio features are presented in Figure 4.3(a) for the Basson sound, Figure 4.3(b) for the Flute, and Figure 4.3(c) and Figure 4.3(d) for the Horn and Principal sounds respectively. The Basson and Horn sound have similar amount of odd and even harmonics where as the Flute and Principal lack even harmonics. In theory, stopped pipes (closed at one end, open at the other), have predominantly odd harmonics. The Basson and Flute have higher Odd contents than Even contents, and these correlate well with the indications given by their harmonics distribution.

MATLAB script to extract modified Tristimulus parameters, odd/even audio features is named *trisOddEven.m*.

4.3.5 Frequency Envelopes and Modulations

In very similar way as for the amplitude envelopes and modulations, the Phase Vocoder analysis also provides information about the frequency envelopes and modulations for each harmonic. These are considered as audio features with an overall view, a more local interest on the first few harmonics with their mimimal/mean/maximal values along the time axis and finally, the frequency deviation as a percentage relative to the harmonic frequency.



(a) Basson (b) Flute

(c) Horn (d) Principal

Figure 4.5: Tristimulus T_1, T_2, T_3, Odd / Even

First, an overall view of the frequency envelopes and modulations is used as shown in Figure 4.6(a). Generally, the frequency modulations are very small for the corresponding harmonics which are predominant (smooth and flat part of the mesh). They correspond to the frequency stability, which in acoustic terms would be the "locking into resonance" at a specific frequency of the harmonic (i.e. definition of the term resonance).

Then, the minimal/mean/maximal values of these harmonics along the time axis are calculated (shown in Figure 4.6(c)), plotted for all the harmonics (Top) and only on a zoomed portion of the lowest harmonics range (Bottom). When the minimal maximal range is very high the frequency envelope can be considered as noise (for which the Phase Vocoder cannot provide accurate results). When this variation is smaller, it can be assumed to be a "real" frequency variation and should be taken into consideration at

the synthesis stage.

Finally, the frequency modulation as a percentage relative to the harmonic frequency is calculated as shown in Figure 4.6(e). This audio feature is useful because it gives some indications about the amount of frequency modulation required at the synthesis stage, and directly relates to amount of noisiness required by the Musicom ADE (MUSE randomisation parameters).

Figure 4.6(a), Figure 4.6(c) and Figure 4.6(d) show these audio features for the Basson sound, while Figure 4.6(b), Figure 4.6(d) and Figure 4.6(f) show the frequency analysis results of the Flute sound. The audio features for the Horn and Principal sounds are shown in Figures 4.7(a), 4.7(c) and 4.7(e), and Figures 4.7(b), 4.7(d) and 4.7(f) respectively.

The audio experts emphasised the importance of frequency modulations to obtain high quality sound synthesis. Both amplitude and frequency modulations are the key audio feature for re-synthesis of high quality sounds. A small amount of randomness is often added to the envelope to achieve this. A statistical study of the spectral parameters in musical instruments tones describing these aspects can be found in [6].

4.3.6 Pitch

Pitch is defined by the ANSI* as *that attribute of auditory sensation in terms of which sounds may be ordered on a musical scale.* To extract the pitch audio feature temporal information from the signal was used to estimate periodicity in the signal. The usual method for deciding if a signal is periodic and then estimating its period is the auto-correlation function:

$$acg(\tau) = \sum_{n=1}^{N} x(t)x(t-\tau) \tag{4.3}$$

*American National Standards Institute

Essentially, the signal x(t) is being convolved with a time-lagged version of itself. To obtain a useful set of results, the auto-correlation function is computed over a range of lag values. It is an important property of the auto-correlation function that it is itself periodic. For periodic signals the function has a maximum at sample lags of 0, ± P, ± 2P, etc. where P is the period of the signal. The sound analysis engine, within WinPVoc, is based on auto-correlation technique. It reads block of sound data from the middle of the sound, performs an auto-correlation and find a maximal value, if a local minimum is found within less than 10% of the maximum value, it considers it as the fundamental frequency. With a size window of 8192 samples it gives an accuracy within the 11 Hz / 8000 Hz range which is good enough for most pipe organ sounds (the range of a pipe organ is from about 16 Hz to 5000 Hz, and the highest fundamental note of the flute is about 4000 Hz). When WinPVoc occasionally fails to estimate accurately the fundamental frequency of the sound, a warning message is returned and other alternative methods should used. The audio experts often use other information like the filename which should give a fundamental frequency using Table A.1) or functionalities of more advanced tools[‡].

A technique for spectral separation described in [109] was implemented in WinPVoc to supplement the original auto-correlation method. Details are given in the WinPVoc manual [58]. More information on pitch can be found in [180] (Chapter 5 dedicated to "Pitch and Pitch Strength"), description of advanced pitch estimation algorithm for speech and music developed in [157] and with a complete review on comparative performance of several pitch detection algorithms in [132].

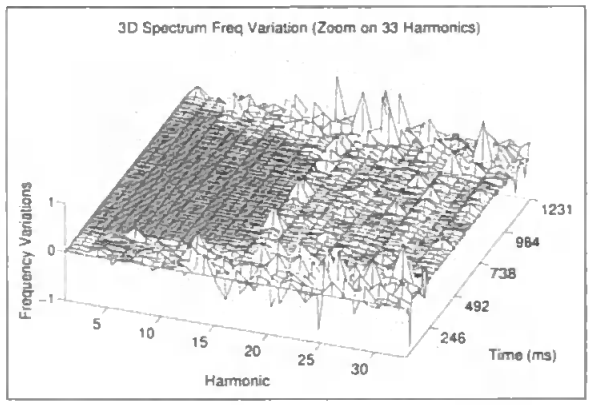
Under the assumption that pitch is a weighted average of estimated fundamental frequencies based on individual harmonics (3 first are considered most effective), it can

[‡]Sound Forge Studio 6.0

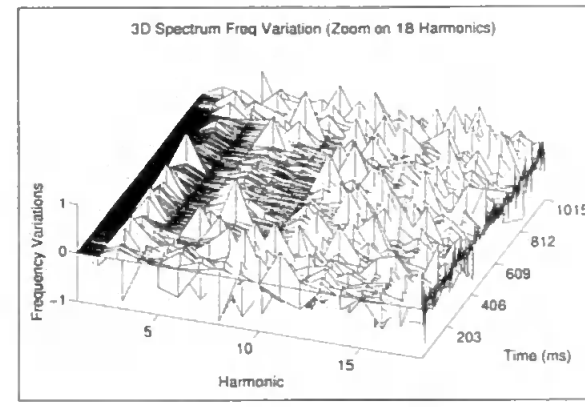
be calculated as follow:

$$Pitch = \frac{\sum_{n=N1}^{N2} A_n F_n}{\sum_{n=N1}^{N2} A_n} \tag{4.4}$$

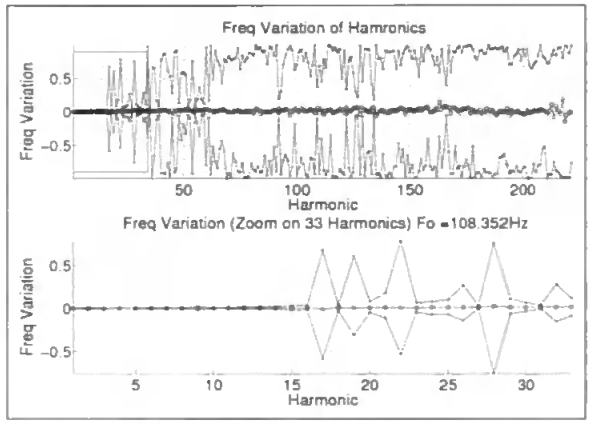
where A_n is the amplitude of the harmonic, F_n is the frequency value, with N1 and N2 indexes usually from 1 to 3 or 2 to 4.



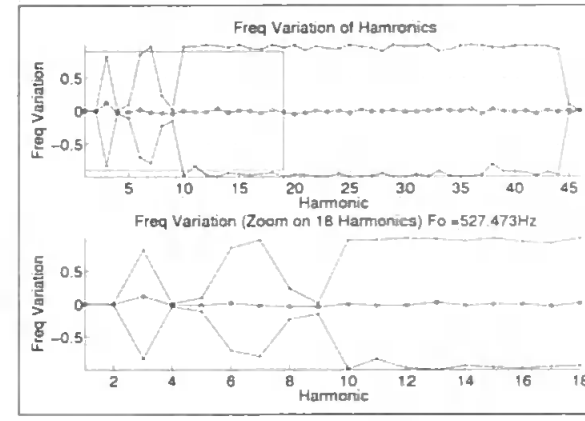
(a) Basson - FreqVarMesh



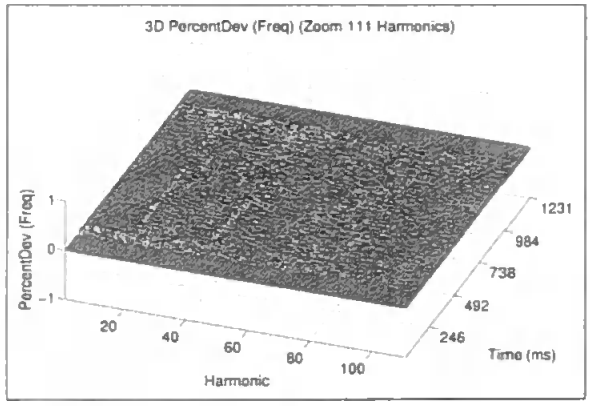
(b) Flute - FreqVarMesh



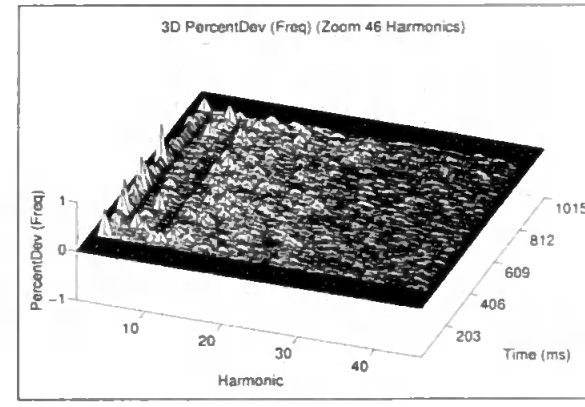
(c) Basson - FreqScatter



(d) Flute - FreqScatter

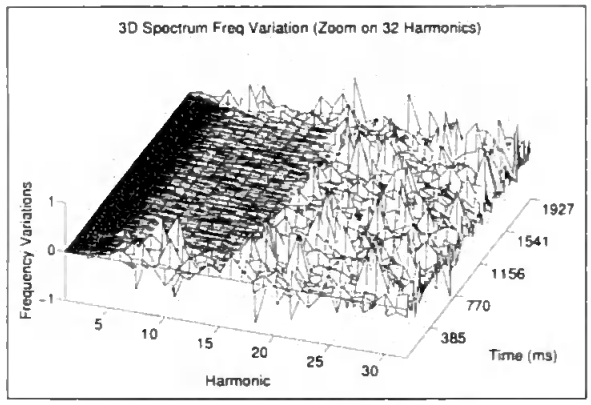


(e) Basson - Percentage Deviation Frequency

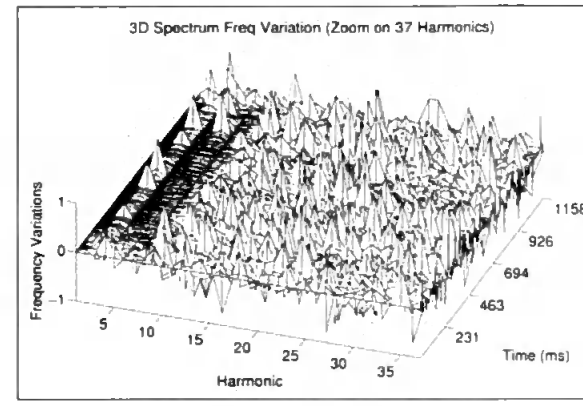


(f) Flute - Percentage Deviation Frequency

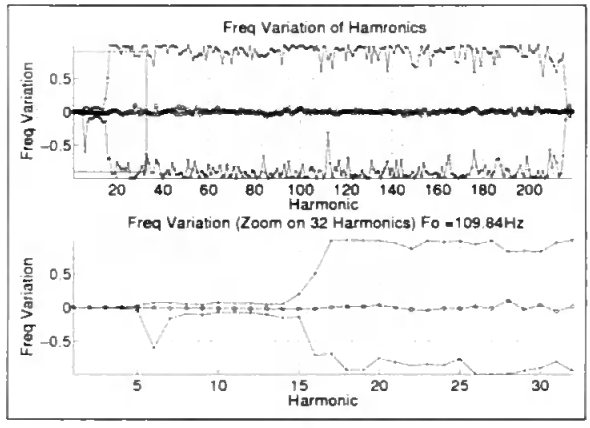
Figure 4.6: Frequency Envelopes and Modulations (Basson and Flute)



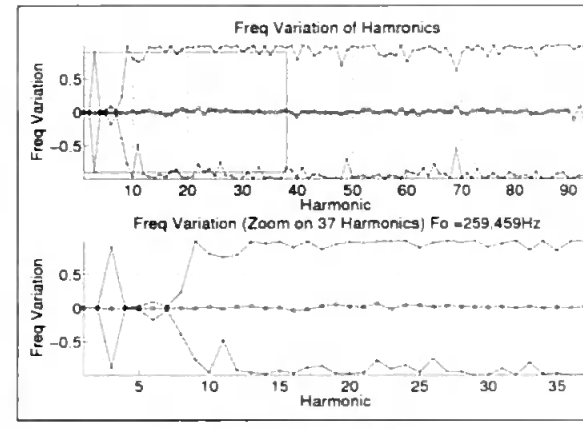
(a) Horn - FreqVarMesh



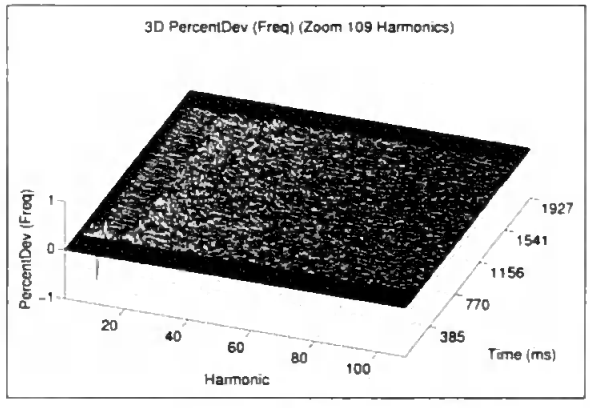
(b) Principal - FreqVarMesh



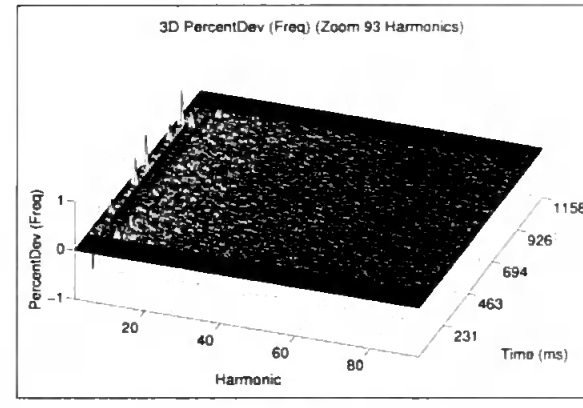
(c) Horn - FreqScatter



(d) Principal - FreqScatter

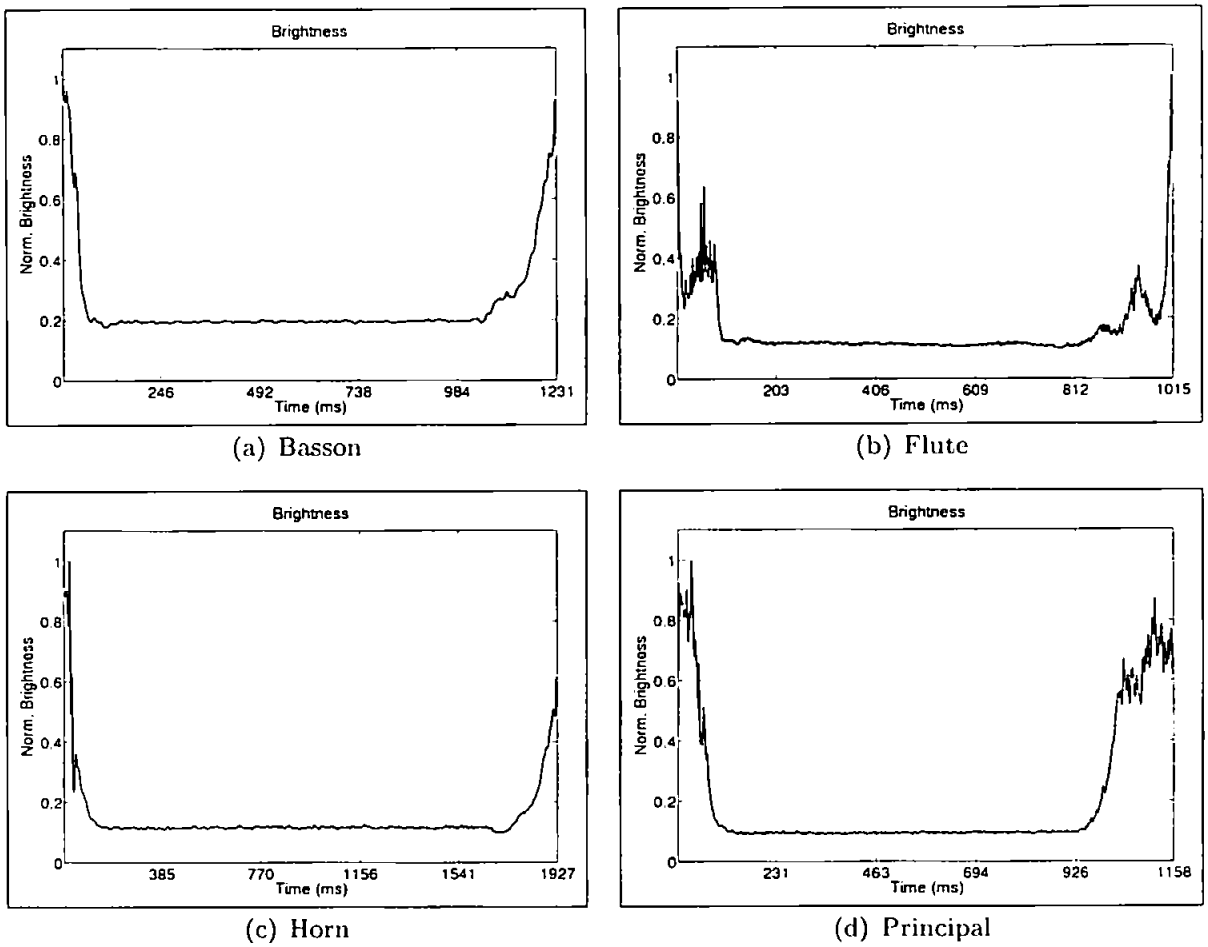


(e) Horn - Percentage Deviation Frequency



(f) Principal - Percentage Deviation Frequency

Figure 4.7: Frequency Envelopes and Modulations (Horn and Principal)



(a) Basson (b) Flute

(c) Horn (d) Principal

Figure 4.8: Brightness of pipe organ sounds

4.3.7 Brightness

The spectral centroid, mathematically defined in Equation 4.5, is a measure of the *brightness* of a sound. It has been found extremely important in the way we compare different sounds, i.e. if two sounds have a radically different centroid, they are generally perceived to be distant in term of timbre.

$$\text{Brightness} = \frac{\sum_{k=1}^{N} k\ A_k}{\sum_{k=1}^{N} A_k} \tag{4.5}$$

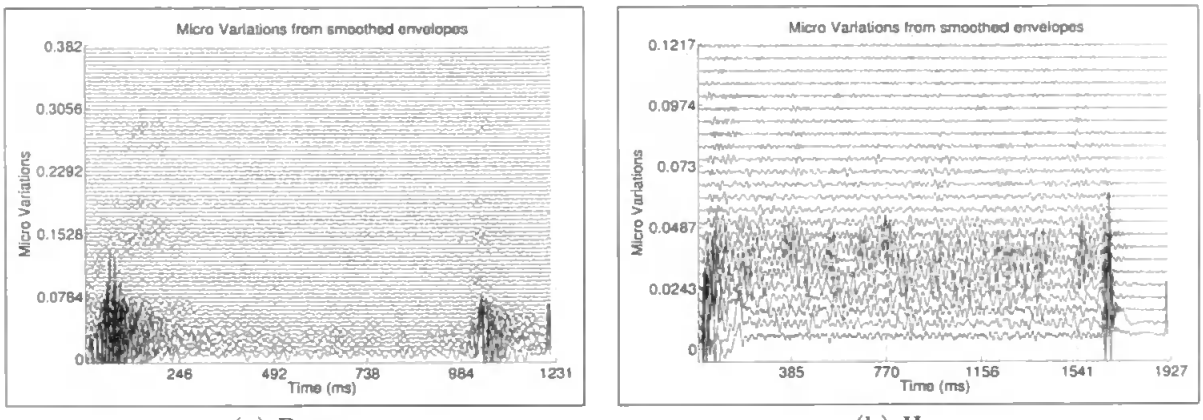
Figure 4.8 illustrates the brightness audio feature. The Flute and Principal sound (Figure 4.8(b) and Figure 4.8(d) respectively) brightness level is constant, with variations corresponding to their breathy characteristic in their attack and release portions (transient effects). All four cases have stable brightness in the sustain portion with Basson sound being "brighter" than the others.

4.3.8 Noise Factor

The noise and randomness issue in sound synthesis is probably the most important issue to give the sound realism. By nature pipe organ sounds are created by acoustic phenomena of air flow against the mouth of the pipe and resonance modes of the pipe itself. Studies of the jet formation [162] and acoustics of pipes [94] are trying to elaborate models to approach this complexity. In sound synthesis, the easiest way is to add some randomisation to the amplitude and frequency envelopes at the re-synthesis stage. Figure 4.9(a) and 4.9(b) show the noise factor audio feature for the Basson and Horn sounds respectively.

This is one of the main reasons why the audio experts do not opted for sampling as synthesis technique, as it often fails to recreate this natural randomness present in any acoustic musical instruments. Other physical aspects, such as temperature variation for example, affect the physical attributes of metal pipes and thus the final sound they produce [71].

An audio feature called noise factor was created, based on the *very small variations* between the data provided by the Phase Vocoder and some very finely smoothed harmonics envelopes. These correspond to the noisiness that is added to the approximated envelopes at the re-synthesis stage. For the frequency aspect, this corresponds to the percentage frequency deviation as shown in Figures 4.7.



(a) Basson (b) Horn

Figure 4.9: Noise Factor of Basson and Horn sounds

4.4 MATLAB Implementation

MATLAB scripts are automatically generated by sound analysis engine after the Phase Vocoder analysis. Each audio feature is associated with a specific MATLAB script, used to extract and plot the features as figures which can be saved as images (JPEG, EPS, etc.) for later use in documents. Some of the developed MATLAB scripts also generate LaTeX files included automatically into reports [57].

Updates versions of WinPVoc (latest version is 0.490) have been regularly emailed to the audio experts who provided valuable feedback used in return to improve the sound analysis engine, discuss the definition and improve techniques to extract audio features. WinPVoc combined with MATLAB provides the audio experts with better sound analysis support, helping them into their work of deeper understanding of pipe organs modeling issues, and most importantly, reduces their day-to-day workload. More information on WinPVoc can be found in its user guide [58].

4.5 Summary

This chapter presented the sound analysis engine and the audio features. The audio features have been defined, methods to extract them detailed and examples given as

illustrations for four different type of pipe organ sounds (namely a Basson, Flute, Horn, and Principal sound). These audio features correspond to key audio features what the audio expert considers in the sound analysis. The sound analysis engine is the front-end of the intelligent audio system, and thus, it is important to extract these audio features as accurately as possible. They are used by the audio feature processing engine, a fuzzy model of audio expertise, to generate optimal sound synthesis parameters. The development of the fuzzy model is described in the next chapter.

Chapter 5

Intelligent System for Audio Features Processing

5.1 Introduction

This chapter presents the audio features processing engine of intelligent audio system as described in Chapter 3. This audio features processing engine is an intelligent system, based on a Fuzzy Expert System (FES) or *fuzzy model* which has been designed, developed and being evaluated in close collaboration with the two audio experts of Musicom Ltd. This *fuzzy model of audio expertise* represents the audio experts skills and knowledge used for the complex task of sound design. The implementation of the fuzzy model of audio expertise is also described.

5.2 Intelligent System

An intelligent system can be defined as a system whose goal is to simulate one or more forms of natural intelligence, for example, learning knowledge and skills, expert behavior, adaptive and evolutionary strategies. The 4 main intelligent techniques are the following: Neural Networks (NNs) [64] that learn to solve problems from examples provided by the human experts, Genetic Algorithms (GAs) [68] that use principles of natural selection and genetics to solve problems with large solution space, Fuzzy Logic (FL) [175] that is

concerned with representing and manipulating information using natural linguistic terms, and finally Expert Systems (ESs) [?] that contain expert knowledge and solve specific real world problems.

Interaction between these intelligent methods, defined above, are often seen to exploit combined features. For example, Fuzzy Logic is often used in expert systems to handle uncertainty and imprecision in the knowledge and data. In Neuro-Fuzzy systems, neural networks are used to learn hidden patterns within the data in order to generate membership functions for the Fuzzy Logic. Expert systems are combined with neural networks to provide them with some explanation capability. Simulated Annealing (SA) [2] are used to tune fuzzy models [43]. It has been now recognised that the full potential of any of these intelligent methods is only likely to be fully realised when they are used in conjunction with other techniques. The development cycle of a fuzzy expert system is now described, introducing the basics of Fuzzy Logic.

5.3 Development Cycle of a Fuzzy Expert System

Figure 5.1 shows a conceptual diagram of a generic fuzzy expert system. Real-world data from the inputs first go through a process of fuzzification. Fuzzification is the process by which real-world data are converted into fuzzy variables using fuzzy sets. The fuzzy inference engine processes these inputs to generate outputs both in form of fuzzy variables which are converted to real-world data after a defuzzification process. In the case of the audio feature processing engine, developed in this work, the real-world inputs are the audio features extracted by the sound analysis engine (See block 1 of Figure 3.1) and real-world outputs are the sound synthesis parameters used by the sound synthesis engine (See block 3 of Figure 3.1).

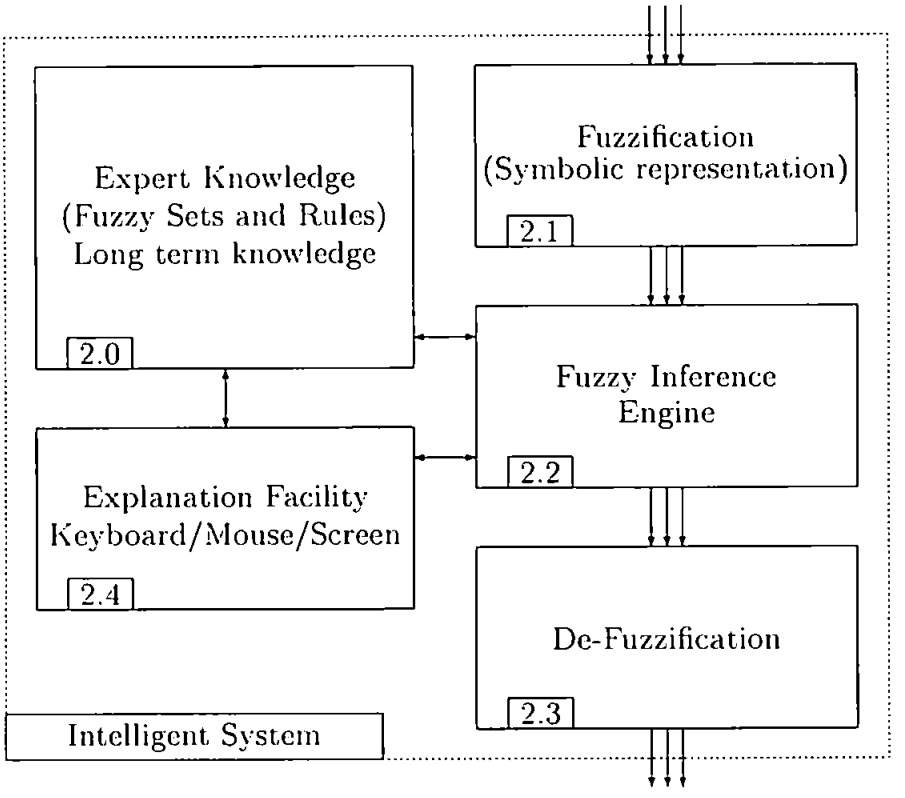


Figure 5.1: Diagram of the Fuzzy Expert System

5.3.1 Basics of Fuzzy Logic

The next sections give a brief introduction to Fuzzy Logic with emphasis on the important issue of *imprecision and uncertainty*. Description of the fuzzy variables follows in Section 5.3.3, fuzzy sets and fuzzy inference process in Section 5.3.4 and Section 5.3.5 respectively. More information can be found in [25] and [175]

5.3.2 Imprecision and Uncertainty

It is necessary to address the fundamental problems of *imprecision and uncertainty* in the computational intelligence model of audio expertise for processing the audio features extracted by the sound analysis engine. Fuzzy Logic is proposed as it offers a comprehensive and flexible framework for handling the imprecision and uncertainty that characterise audio knowledge, data and decision-making [176]. In particular, it provides a framework for

describing, manipulating, conveying information and drawing conclusions using linguistic terms as in the real world. Thus, with Fuzzy Logic, the linguistic terms (or variables) and rules which the music technologists use and understand can be used directly. This makes the model accessible to music technologists/audio experts in a natural form and which is an important factor in the successful development of the model.

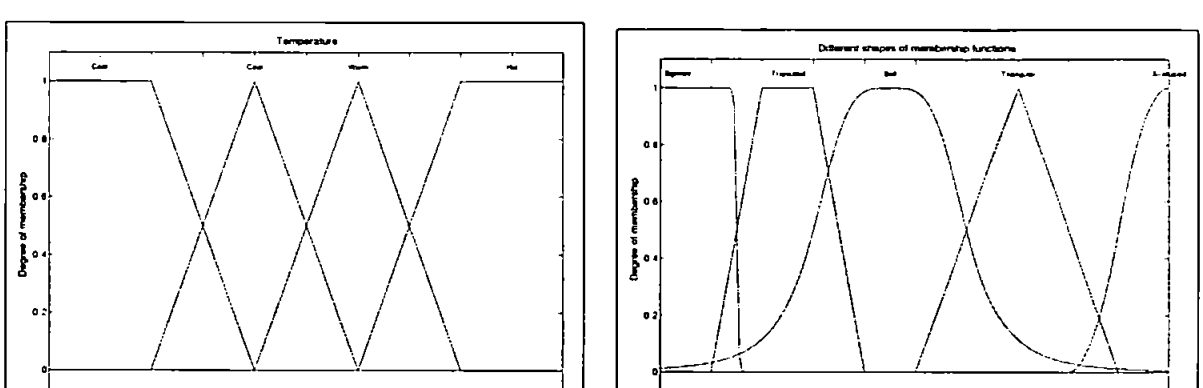
A feature of Fuzzy Logic is that the use of the same natural language terms as human experts greatly simplifies the design of complex systems as it obviates the need for complex mathematics. This attribute has inspired many practical applications, especially in control and consumer electronics for example auto focus in cameras [33] and medical systems [77][90]. Fuzzy Logic is based on two key concepts, namely linguistic variables and fuzzy sets.

5.3.3 Fuzzy Variables

Linguistic variables (also called fuzzy variables) are subjective, context-dependent variables whose values are words. For example, if the word temperature is regarded as a linguistic variable, its values could be cold, cool, warm and hot and can be denoted as - temperature (cold, cool, warm, hot) (see Figure 5.2(a)). The use of linguistic variables considerably simplifies rules in fuzzy systems as it makes possible to employ the same language used by human experts and to develop a natural rule set. Fuzzy sets are used to represent linguistic variables.

5.3.4 Fuzzy Sets

A fuzzy set denotes a class of elements of a variable with loosely defined boundaries. The grade of membership (which lies between 0 and 1) indicates the degree of belonging of an element to a fuzzy set. Each fuzzy set is represented by a function (e.g. trapeziodal, bell, sigmoid, triangular, s-shaped as shown in figure 5.2(b)) which specifies how



(a) Fuzzy sets for temperature (b) Different shapes of membership functions

Figure 5.2: Fuzzy sets and membership functions

the membership in the set is distributed. The grades of membership may be initially assigned subjectively by human experts or from analysis of the problem domain. Thus, the development of a fuzzy model of expertise involves determining a set of linguistic variables, the corresponding fuzzy sets with membership values for each set, and formulating a set of fuzzy rules that describe the input-output mappings. The fuzzy output sets may be defuzzified to produce an index which could be used, for example, to control sound synthesis engine. The model may be tuned to optimise performance relative to an objective measure.

5.3.5 Fuzzy Inference

To apply the model involves three basic operations. First, the input variables, e.g. sound features, must be converted into fuzzy variables using the membership functions. Then a fuzzy inference engine processes the variables by applying the rules. Essentially, for each rule in turn, the membership grade for each fuzzy set is evaluated by determining the value of membership function at the given value of the corresponding input variable. Membership grades are then combined by a suitable fuzzy operators (e.g. min and max operators) to give the extent the rule will be activated. This information is used to

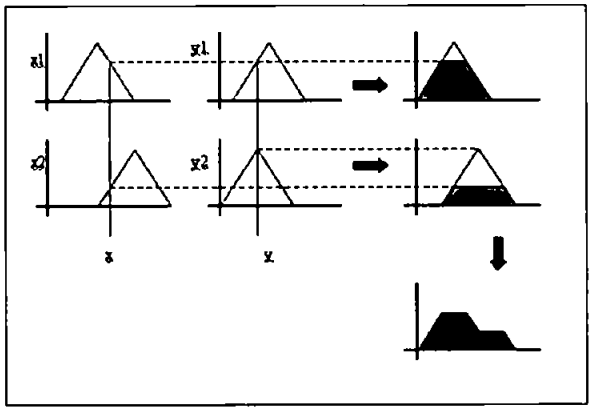


Figure 5.3: Fuzzy Inference

truncate the associated rule consequent fuzzy set. The consequences of all rules that activated are combined to give an overall fuzzy consequent. The fuzzy consequence is then usually defuzzified by finding its centroid to produce a crisp output for the outside world (in our case sound synthesis parameters to control synthesis engine). The process for a two rule system is illustrated in Figure 5.3.

5.4 Development cycle of the Fuzzy Model of Audio Expertise

The development of a fuzzy expert system is a long and recursive process. At each stage a feedback path can be allocated to if the results are not satisfactory. The fuzzy model was developed with the guidance of Tony Koorlander and Graham Blyth both from Musicom Ltd. The development cycle for the audio expertise model is shown in Figure 5.4

The development of a typical fuzzy expert system can be divided into the following stages, each stage involving some tuning of its own. Figure 5.4 shows the development cycle process. The development cycle starts with the definition of the problem that the audio experts face during the sound design process and to a wider extent the full modeling of musical instruments such as an pipe organ. Knowledge elicitation sessions are carried

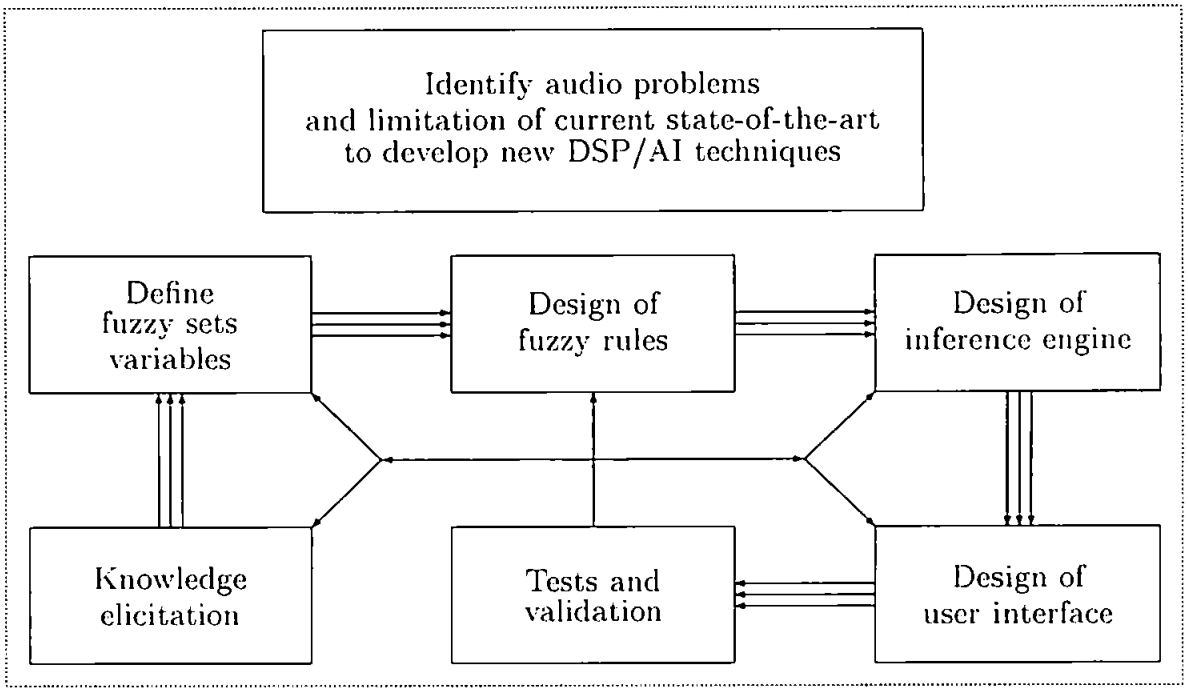


Figure 5.4: Development cycle of a typical Fuzzy Expert System

out to define the task of sound design and identify the requirements about the task. All this was followed with the definition of the fuzzy sets and fuzzy variables used in the fuzzification (Input fuzzy variables) and defuzzification process (Output fuzzy variables). The design of the rules followed together with the design of the fuzzy inference engine, basis of the rule-base system, to finish with the user interface (using Matlab GUI) and tests and validation. This development cycle aimed at identifying real audio problems related to the one described in Section 1.2, (i.e. the limitations of the current state-of-the-art) and at developing new techniques to overcome these current limitations. The developments of the *fuzzy model of audio expertise* are described after the next sections on knowledge elicitation.

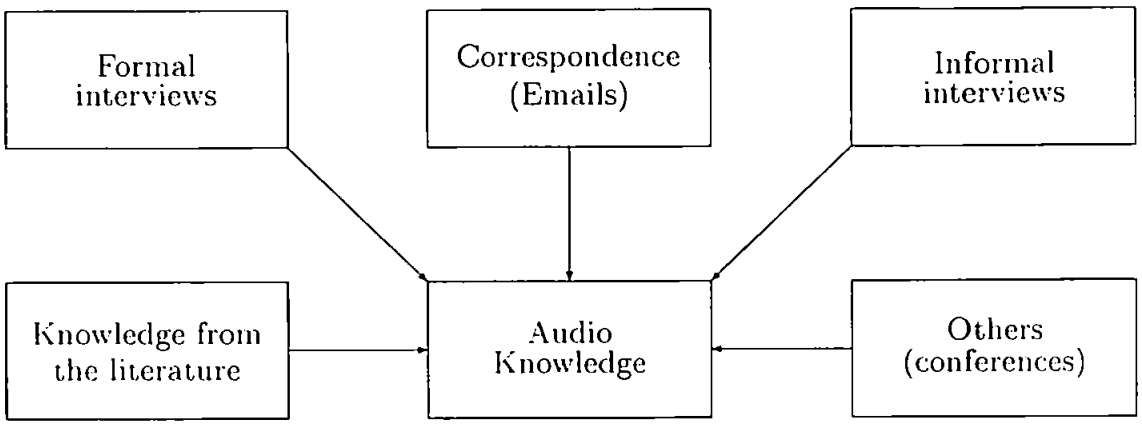


Figure 5.5: Knowledge elicitation sources

5.5 Knowledge Elicitation

Knowledge elicitation is the process by which the expert knowledge is collected and formulated to be used for the development of the fuzzy model. Over the all course of this research project, knowledge elicitation was continuously conducted and reviewed with the audio experts. Audio knowledge was collected from different sources to help develop the *fuzzy model of audio expertise*. Different elicitation techniques, as shown in Figure 5.5, have been used to obtain sufficient audio knowledge necessary for the the design, development of the fuzzy model. These include knowledge from the literature, formal interviews, correspondence by emails, informal interviews, others source like conferences. They are now discussed in more detail.

5.5.1 Knowledge from the literature

The initial investigation into modeling musical instruments and sound design was to conduct a literature survey, with particular interest in the use of Artificial Intelligence techniques in the field of audio. Key publications from sources such as IEEE Transactions on Speech and Audio Processing, on Neural Networks, on Fuzzy Systems and on Evolutionary Computation. Journal of the Audio Engineering Society (JAES) and Computer Music Journal (CMJ) publications were also collected and reviewed. Same process has

been carried out with conferences publications from proceedings of signal processing and audio conferences such as International Conference on Acoustics Speech and Signal Processing (ICASSP), International Conference on Digital Audio Effects (DAFX), Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), International Computer Music Conference (ICMC). Without audio and musical training as well as reading journal articles, it would have been very difficult to assess the size of the task of this research project. The author's formation is in electronic engineering with a artificial Intelligence (AI). The author would like to emphasis that this research work has been conducted with basic musical background.

If signal processing conferences and journals publications with emphasis on sound synthesis and instrument modeling have been valuable sources of information, discussions with the audio experts gave even more interesting information with often a practical view, looking beyond the academic and theoretical issues. It has been recognised that there was important gap between the theoretical issues and practical ones. Musicom Ltd audio experts being faced with both.

The literature review provided a lot of information about the pipe organ acoustics, that helped the author to gain a better understanding about the instrument and initial audio expertise. The acoustics of pipe organs has been at the center of attention of many researchers including Hirschberg and Verge (IRCAM). This vast expertise on the acoustics of pipe organ available from the literature does not always serve the purpose of electronic musical instrument modeling but rather investigation in the acoustics and from a physical point of view [94][63]. This has been used as background information for the research. Furthermore, information provided by Musicom Ltd about the ADE system [100][99] was also very helpful, providing a clear picture of the current state-of-the-art in the electronic pipe organ industry. Comerford (Bradford University, UK) was the pioneer in modeling pipe organs using electronic systems [21][22]. The *Bradford Musical Instrument Simulator* (BMIS) [23] system can be considered as the basis of the early Musicom ADE systems.

Over the years, Musicom Ltd has developed, independently from Comerford's research, further the concept. The results are now that Musicom's ADE system is now recognised as one of the best electronic pipe organ system and used to replace acoustic pipe organ all over the world [97][98].

The literature survey, especially related to psychoacoustics, provided interesting topics for many discussions with the audio experts. It has been found that more and more researchers are taking advantages of psychoacoustics to improve their research. Horner, for example, recently augmented his research with a perceptual flavor for his Genetic Algorithms (GAs) based optimisation of wavetable sound synthesis [169].

It is interesting to notice that in many aspects of the audio experts work, are based on psychoacoustics. The typical *masking effect* of higher harmonics is one clear example of this [106]. Finally, over the course of this research project, it has been found that psychoacoustic considerations are the helping aspects of this research project and that issues of *perceived sound quality* will be the motivation of many future audio research topics (See sections 7.5).

5.5.2 Formal interviews

Formal interviews were conducted to elicit information from Tony Koorlander and Graham Blyth from Musicom Ltd (Bideford, Devon), both regarded as authorities in electronic pipe organ pipe sound synthesis. Musicom Ltd is a small company but provides very high quality products, mostly because the founders of the company are passionate as electronic engineers (design and development of the electronic instrument) as well as musicians themselves (users of acoustic and electronic musical instruments). Many visits to Musicom facilities have been organised. Meeting were conducted with a structured agenda and often hand-on demonstrations. Many demonstrations of sound design process have been given by the audio experts. These allowed to gain experience into the complex process of sound design and to identify important issues involved. Many aspects of this

research project became clearer after that the audio experts have shown, often repeated many times, how they tackle the task of sound design. The author must admit here that his lack of musical knowledge has not been too problematic but would have certainly facilitated the research.

The major disadvantage of formal interview is that without prior knowledge it was difficult for the author to elaborate questions. A major difficulty in this research project was that much of the *audio knowledge* is not usually strictly defined, with much of the audio experts work based on the case to case basis (client basis would even be a more appropriate term). An important overview was obtained from the formal interviews, but it was often felt at the time, that alone, they were still restrictive. Some of these meetings have been recorded and transcribed.

5.5.3 Informal interviews

Most of the audio and sound design understanding came from many hours of informal interviews and discussions which were conducted throughout the period of this research project. Equally important was that it was a way to keep the busy audio experts closely involved in the development of the system. Audio experts, like Tony Koorlander and Graham Blyth, do not have much free time nor the necessary Digital Signal Processing (DSP) and Artificial Intelligence (AI) background to overview the research process in detail, but it was important that they help the sharing of audio knowledge. The audio experts have more than 15 years experience working in the field of development, design and use of electronic musical instruments and this make them the *center of expertise* to help develop such intelligent system described in this thesis. Finally, informal interviews often acted as a motivating factor which ensured that the developments were focused. Many novel ideas were born, have evolved and been clarified as the results of these fruitful discussions.

5.5.4 Correspondence

It has been necessary to use electronic correspondence with the audio experts to continuously refine the knowledge used in the system. Audio experts are rarely accessible, but with the easy and low cost access to electronic mail, access to their expertise had been greatly facilitated. Emails have been used extensively to communicate ideas, ask questions, discuss and elaborate and explore potential solutions with both research and technical point of views. It has to be said that the day-to-day work and business activities of audio experts do not allow continuous contribution and instant feedback. The development of an fuzzy expert system, like the one described in this thesis required a lot of time, attention to details and would ideally requires a fast communication channel with the audio experts. Feedback from the audio experts was, however, always the most valuable. Updated version of the software tool WinPVoc was regularly sent to the audio experts for evaluation. Experts always gave feedback with sometimes request for features to be modify or added, improvements to be made and suggestions for research.

5.5.5 Other researchers and conferences

Information has also been collected from other sources such as audio engineers and researchers at conferences. During the course of this project two conferences have been attended, both having stimulated many novel ideas, looking at other approach and techniques. During visits to Musicom Ltd there have been many occasions to discuss about the project and research with other engineers at Musicom, both in the hardware and software aspects.

It has also been useful to visit local churches and request a demonstration (when possible) of the acoustic pipe organ. Enquiries to other electronic pipe organ manufacturers such as Allen Organ Company[†] or Phoenix Organs[‡] for brochures and audio CDs provided

[†] `http://www.allenorgan.com`
[‡] `http://www.phoenixorgans.co.uk`

some information about Musicom competitors as well as the current state-of-the-art.

Internet websites, like Rodgers Instruments [138], use also many *marketing buzz words* like Voice Palette™ technology, Organ Designer® software, Digital Dynamic Wind™, Digital Domain Expression™, Random Tuning, Stereo PDI™ sampling, True Chimes™ sampling, Rodgers Intelligent Sampling System™, which are never really detailed in the brochures.

It was found from discussions at the Pipe.org mailing list [127] that most modern electronic pipe organs systems are real-time embedded systems using sampling technology or many low cost wavetable synthesis chips (e.g. from Dream S.A. an Atmel semi-conductor manufacturer). On the other hand, Musicom uses technology based on early Comerford patent [21] with many years of additional and independent developments (based on customer's feedback).Finally, Musicom uses their own silicon solution (ASIC), showing real advancement in electronic pipe organ technologies.

5.5.6 Discussion

The author has found the process of knowledge elicitation extremely enriching and working with audio experts, has added a practical aspect to the often very theoretical academic research in audio signal processing. The audio experts often looked at audio issues with an industry's point of view i.e. including both practical and commercial aspects, which contributed greatly to the progress of the project.

5.6 Design of Fuzzy Sets and Variables

The following sections describe the design process used to develop the fuzzy sets and variables of the fuzzy model inputs and outputs. Each fuzzy variable makes use of the corresponding audio features extracted by the sound analysis engine. The rationale behind the design of each fuzzy set and variable is detailed.

It should be emphasised that the work presented in this thesis is a first in computer music attempt to the development of an intelligent system that encapsulates audio expertise into a fuzzy computational model to support the sound design process. This unique audio research departs from any other work published to date in audio research. Other related work in other domains such as medical [44][45] helped to inspire the present development as there is no audio research describing such development of a fuzzy model of audio expertise.

The definition of the fuzzy sets and rules with the audio experts has been a long process. The optimisation loop, described in Section 3.6, helped to automatically tune the fuzzy model within an iterative loop. This optimisation is based on the perceptual audio quality of the sounds the system can produce.

Design choices to be made at the initial development stage included the number of variables and sets, the type of membership function, and inference methodology. More advanced fuzzy model also include modifiers, operators and edges [25].

In the fuzzy model, sigmoid type of membership functions were used for all the fuzzy sets, and it was decided (preferred) to limit the number of fuzzy sets per variable (with few exceptions). Sigmoid membership function gives a smooth transition across the fuzzy sets along the fuzzy variable's *universe of discourse* and can be adjusted using a parameter [139]. Variables with only few fuzzy sets was a choice to try not to complicate too much the initial design of the fuzzy model, when discussing the choice of the fuzzy sets (crossing points). This can be easily extended with more fuzzy sets (and more variables), as the development of the fuzzy model get further evaluated by the audio experts.

No claims are made as which type of membership function is best in the context of this fuzzy model. The initial choice of sigmoid-shaped membership function was made because of possible adjustable smoothness parameter (See variable N in Appendix B.3 and B.5 and in [139]). The effect of alternative membership functions such as trapezoid, triangular, S-shape, etc. (See Figure 5.2(b)) is left to future work (See Section 7.5).

The *universe of discourse* of all the fuzzy variables ranges between 0 to 100 (with one exception). This was chosen to ease the understanding by the audio expert (i.e. looking at audio feature in terms of percentage). In the fuzzy model implementation, all the audio features extracted were normalised (to 100 using some scaling factors to adjust at best for each) and this further simplify the development, current and future, of the fuzzy model.

5.7 Fuzzy Inputs from Audio Features

The following sections detail the fuzzy input variables and associated fuzzy sets developed for the fuzzy expert system. The fuzzy inputs include the following variables: Attack, Decay and Release Time, Harmonic, Amplitude and Amplitude Modulation, Frequency and Frequency Modulation, Pitch, Brightness and finally Noise. These fuzzy variables are based on the audio features described in Chapter 4 and extracted using the sound analysis engine.

5.7.1 Input - Attack Time

The design of the fuzzy sets for the Attack Time fuzzy variable has been based on the advice of the audio experts and statistical study of the close inspection of the attack segment of many examples of pipe organ sound analysis. Figure 3.6 showed a typical example of the evolution of the harmonics during the attack portion of a Basson sound. Thick vertical lines were showing estimates of the attack time for each harmonic. It was obvious that some harmonics have slower attack than others.

The Attack Time fuzzy sets consists of a *VeryFast*, *Fast*, *Medium*, *Slow* and *Very Slow* fuzzy set. The fuzzy sets are shown in Figure 5.6(a). If a very very long attack time is considered to be 100 ms, then the points at which the fuzzy sets cross on the *universe of discourse* (horizontal axis) correspond to the following timing of 0 to 3 ms

for a *VeryFast* attack, 3 to 12 ms for a *Fast* attack, an attack time was considered to be *Normal* between 12 and 30 ms, whereas a *Slow* attack would be between 30 and 80 ms, and finally a *VerySlow* attack time would be 80 ms and more. These numerical values correspond to the typical attack values used in the Musicom ADE system.

5.7.2 Input - Decay Time

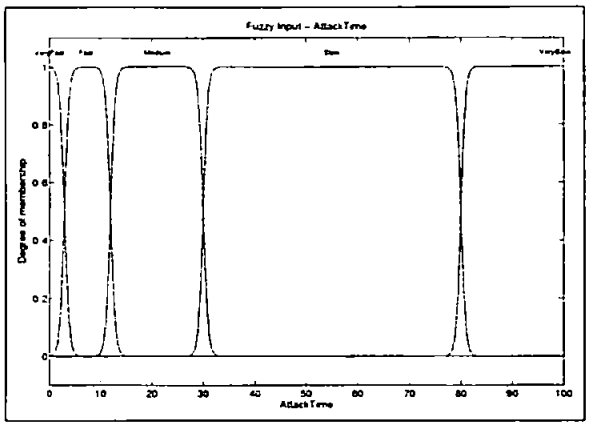
Decay segments do appear in only few typical sound cases. and their importance was not emphasised by the audio experts. As show in Figure 5.6(b) the Decay Time fuzzy variable consists of only 3 fuzzy sets: a *Fast*, *Medium*, and *Slow* Decay Time. The crossing points between the fuzzy sets correspond to 5 ms (*Fast* to *Medium*) and 15 ms (*Medium* to *Slow*).

5.7.3 Input - Release Time

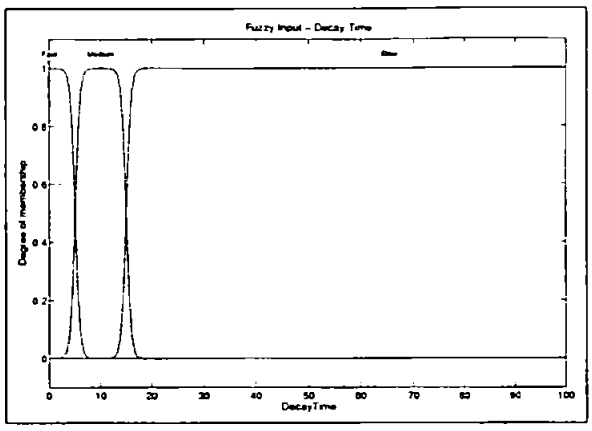
During the sound design process, the release portion of the sound is not an issue considered very important by the audio experts. With pipe organ sounds, it is often dealt with fading out the whole sound using decaying envelopes. The Release Time fuzzy sets used to create the fuzzy variable, shown in Figure 5.6(c), are thus using the similar values as the attack time fuzzy sets. In theory, a difference should be made between attack and release as, from a psychoacoustic point of view, perceptual masking considerations defines that the duration for forward masking are different from the one dealing with backward masking [180].

5.7.4 Input - Harmonic

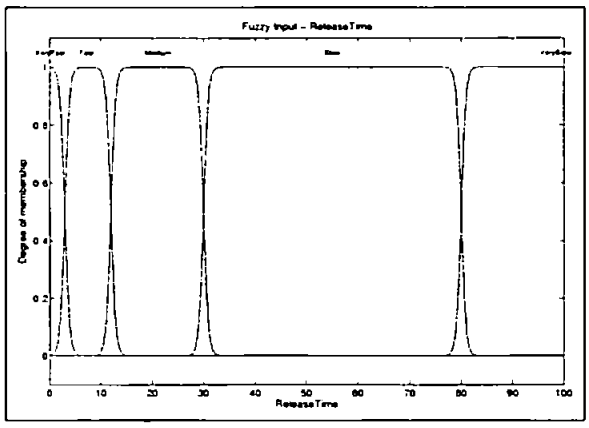
The fourth input of the fuzzy model is the Harmonic fuzzy variable. It is one important input of the fuzzy model as, following the discussion given in Section 4.3.3, it should be clear that this fuzzy variable will be used for assessing individual harmonics in the



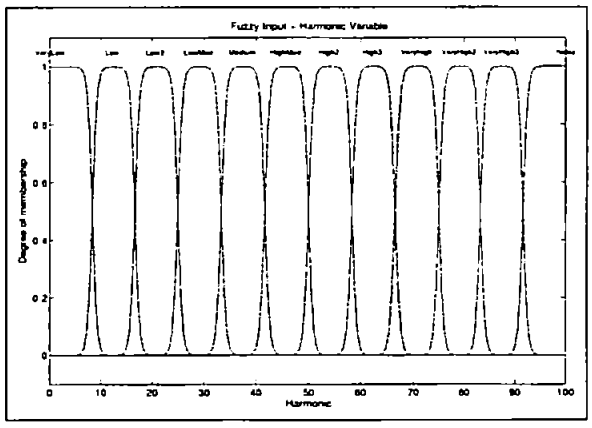
(a) Attack Time



(b) Decay Time



(c) Release Time



(d) Harmonic

Figure 5.6: Fuzzy sets - Inputs Time

sound design process, about their relative position in the sound harmonic distribution, in relation with other closer harmonics, as well as with the same harmonic indexing (odd/even). To obtain an accurate selection of individual harmonic, this fuzzy variable consists of 10 fuzzy sets as shown in Figure 5.6(d). The maximal value of 100 on the *universe of discourse* axis corresponds to harmonic the 12^{th} harmonic. It is worth noticing that for most cases, the current number of fuzzy sets is large enough however for sounds in the very low register (as some pipes have fundamentals of 16 Hz) this fuzzy variable will require a few more sets to be added.

5.7.5 Input - Amplitude

Figure 5.7(a) shows the fuzzy sets used by the fuzzy variable Amplitude, made of 7 fuzzy sets as follows: *VeryLow*, *Low*, *MedLow*, *Medium*, *MedHigh*, *High*, and *VeryHigh*. Most important are the 3 fuzzy sets corresponding to the highest amplitude and the one covering the lowest amplitude, as this fuzzy variable will be used to group harmonics with high amplitude (considered to contribute the most to the sound) and the fuzzy set *VeryLow* for very low amplitude (used in combination with the Noise fuzzy variable).

5.7.6 Input - Frequency

The fuzzy sets used by the Frequency fuzzy variable are shown Figure 5.7(b). This variable is use to assess the reference of each harmonic frequency. It is made of 5 fuzzy sets with the *VeryLow*, *Low*, *Medium* , *High*, and *VeryHigh*. The value 50 (center of fuzzy set *Medium*) corresponds to the center of the frequency envelope.

5.7.7 Input - Amplitude Modulation

Figure 5.7(c) shows the fuzzy sets used by the fuzzy variable Amplitude Modulation, which consists of 6 fuzzy sets named *VeryLow*, *Low*, *MedLow*, *Medium*, *High*, and finally *VeryHigh*. Most important are the 3 fuzzy sets corresponding to the lowest amplitude modulations. It is worth noticing that the fuzzy set *VeryHigh* is (very) rarely used by the system and that the three first fuzzy sets have been designed to make the variable sensitive to small amplitude variations (see proportion of the three first fuzzy sets compared to the total *universe of discourse*). The value maximum of 100 corresponds to an "empirical" maximal variation of 0.35 of the level of an normalised harmonic.

5.7.8 Input - Frequency Modulation

The frequency modulation fuzzy variable is made of 6 fuzzy sets, shown in Figure 5.7(d). Fuzzy sets crossing point correspond to a percentage of frequency deviation *VeryLow* between 0.000 and 0.017, *Low* corresponds to a percentage of frequency deviation between 0.017 and 0.03, *MedLow* corresponds to values between 0.03 and 0.063, *Medium* between 0.063 and 0.128, *High* between 0.128 and 0.25, *VeryHigh* between 0.25 upwards. These crossing points were given by the audio experts referring to WinPVoc software routine used to calculate the Musicom ADE frequency randomisation values. These values have been carefully chosen by the audio experts and represent an accurate multi-level of frequency modulation parameters settings.

5.7.9 Input - Pitch

The pitch audio feature defined in Section 4.3.6 is used as fuzzy variable made as shown in Figure 5.8(a). This fuzzy variable consist of 7 fuzzy sets, named *Very low* to *Very high*. With a maximal value of 108 on its *universe of discourse* axis it correspond to the maximal key note of the Musicom keyboard. Referring to Table A.1 the transition between the fuzzy sets correspond to the keys $C1$ (*VeryLow* to *Low*), and then $C2$ (*Low* to *MedLow*), etc. up to $C6$ (*VeryLow* to *Low*). The fuzzy sets are cross at a higher membership level than the usual 0.5, this is done on purpose to include the $C1$, $C2$, etc. keys membership has more weight. This fuzzy variable will allow the fuzzy model to check from which octave the harmonic being processed belong to. It is known from the audio experts that low pitch sounds require more care in the sound design process than high pitch. The same apply for the resources used, a low register gamba sound requires more resources than a high light flute sound, which only required, generally one signal basic generator.

5.7.10 Input - Brightness

The next fuzzy input variable is the Brightness, based on the audio feature of the same name defined in Section 4.3.7. Figure 5.8(b) shows the fuzzy sets used for the brightness fuzzy variable. This variable is used mostly with high pitched sounds like flute sounds, which usually present a very high brightness.

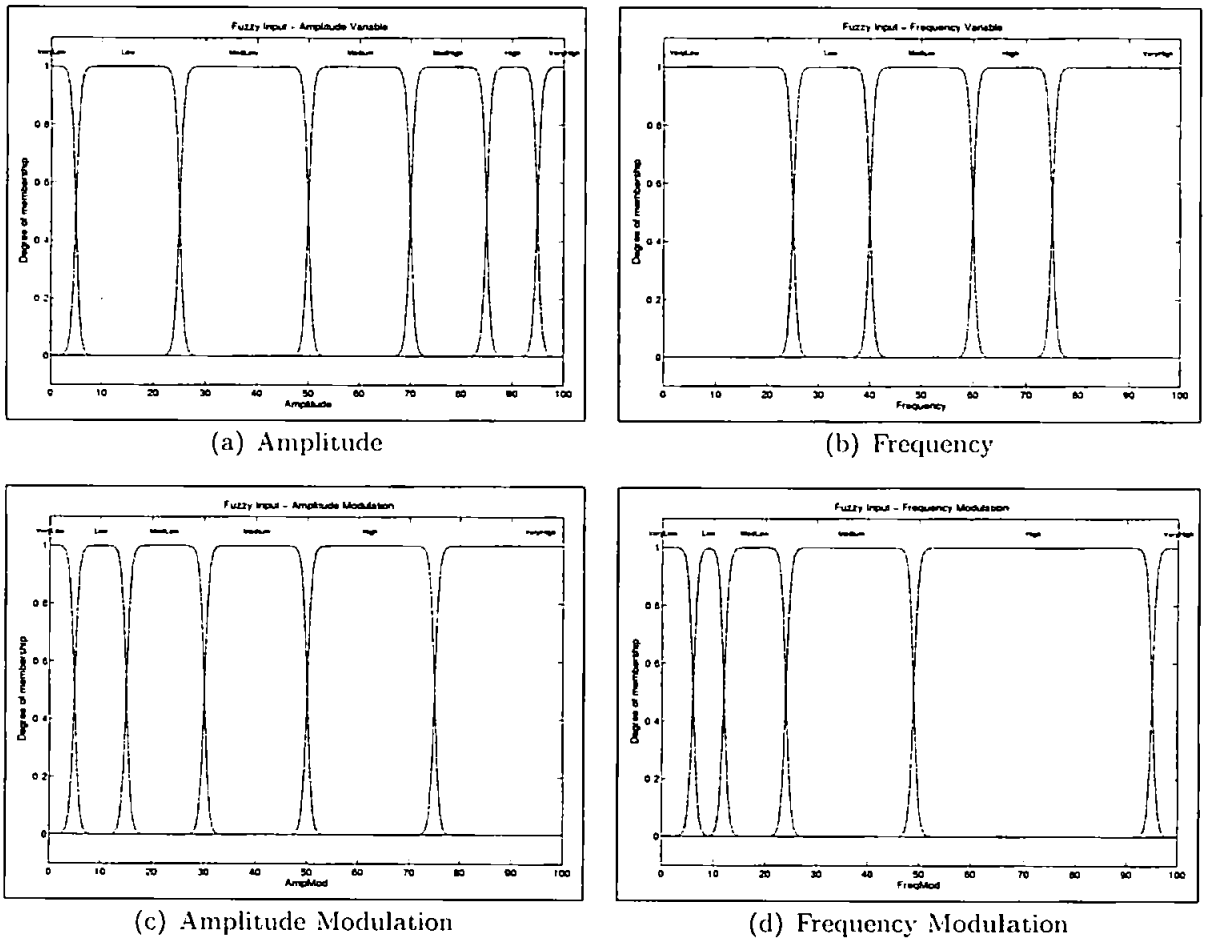
5.7.11 Input - Noise

The final input fuzzy variable is the Noise and is using the results from the sound analysis engine with the audio feature called Noise Factor described in Section 4.3.8. It consists of 5 fuzzy sets which are shown in Figure 5.8(c). Each set are equally divided along the fuzzy variable's *universe of discourse*. The maximum values of 100, extreme values of its *universe of discourse* corresponds to 0.05 (relative to a 1.0 normalised harmonic amplitude). This variable is meant to help represent a more accurate version of the Amplitude Variation variable (with its *VeryLow* fuzzy set), and to deal with the noisiness of each harmonics.

5.7.12 Discussion

The input fuzzy variables and their associated fuzzy sets have been developed in collaboration with the audio experts. The fuzzy variables, defined with linguistic terms understandable by humans, represent the key features that the audio experts are examining when designing pipe organ sounds. These audio features can be considered as *generic* and would most probably apply to other types of musical instruments, with the fuzzy sets adjusted to fit the particular sound characteristics of that instrument.

The definition of the fuzzy sets has been a long and time-consuming process. In future, a large sound database will be used in an attempt to automatically define and fine-tune the fuzzy model parameters using techniques such as Simulated Annealing (SA) [1] which



(a) Amplitude

(b) Frequency

(c) Amplitude Modulation

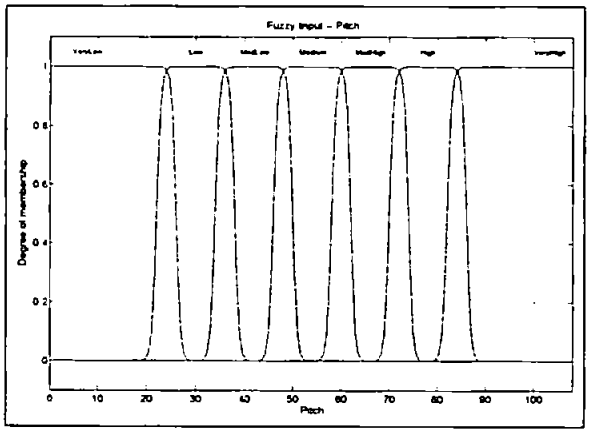
(d) Frequency Modulation

Figure 5.7: Fuzzy sets - Inputs Amp and Freq

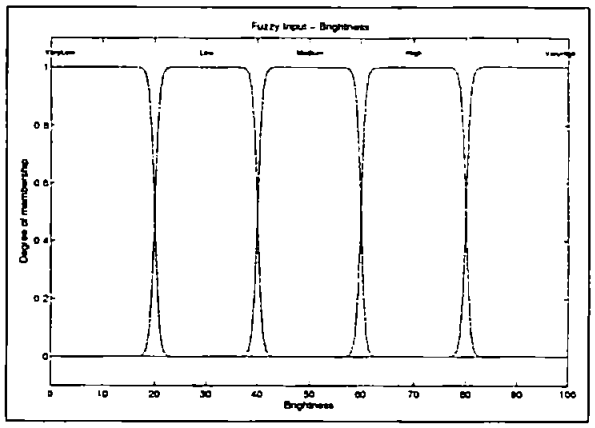
have been shown successful in other fuzzy models [43].

5.8 Fuzzy Outputs to Synthesis Parameters

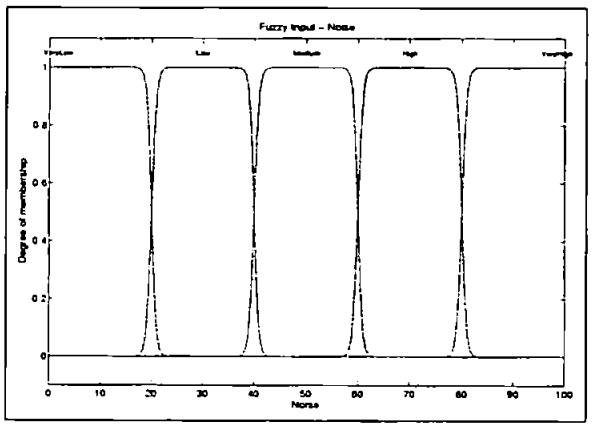
The aim of the audio feature processing engine, implemented as a fuzzy model, is to generate sub-optimal parameters for the sound synthesis engine, using audio expertise to mimic the audio experts work. At the final stage, the fuzzy model needs to transform, by defuzzification process, the fuzzy outputs into real-word data, used to configure and control the sound synthesis engine. Four distinct fuzzy outputs that represent key aspects of the control of the sound synthesis engine have been defined with the audio experts as



(a) Pitch



(b) Brightness



(c) Noise

Figure 5.8: Fuzzy sets - Inputs Pitch, Brightness and Noise

the following:

- **Cluster** - The Cluster fuzzy output provides an indication about which cluster the harmonic should belong to. Like the audio experts working on designing the sound, the fuzzy model tries to assess the contribution of the harmonic to the final sound quality and groups harmonics into clusters, or isolate them from their particularities, for the sound synthesis.

- **Attack Sound Synthesis** - The Attack sound synthesis fuzzy variables include Amplitude Envelopes, Amplitude Modulations, Frequency Envelopes and Frequency Modulations, with the Modulation being further refined with Amount and Rate pa-

rameters. Each of the 6 fuzzy variables has its own fuzzy sets described in the next sections. These parameters are considered important by the audio experts to obtain realistic and high quality pipe organ sounds.

- **Sustain Sound Synthesis** - The sustain sound synthesis parameters also include the same parameters with Envelopes, Modulation with additional Amount and Rate parameters, for both Amplitude and Frequency. Sustain parameters result, however, from a distinct set of sound design rules that the audio expert uses. They correspond to the initial stage of the audio expert decision process, the attack section being examined only once the results of the sustain have been considered satisfactory. Sustain corresponds to a different perceptual stage of the sound, more "stable" compared to the "transients" of the attack .

  Modulations in both amplitude and frequency are issues on which the audio experts spend most of their time, taking care of the small amount of amplitude and frequency modulations, in ways that it adds "life" to the sound in ways that sampling cannot replicate [97].

  At the sound synthesis engine stage these parameters are used to create wavetables made of a static spectrum image (created by the Musicom ADE hardware's IFFT) "brought to life" by the advanced modulation in amplitude with the main enveloped added of some modulations with adjustable amount and also rate at which these modulation are processed by the Musicom ADE. More information can be found in the VASM User Manual [99] from Musicom Ltd.

- **Noise Generator** - The Noise Generator fuzzy variable describes an additional source that adds small amount noisy signal to improve the overall sound realism and to emulate typical pipe organs characteristic such as *chiff* for example. Its fuzzy sets comprise of a *NoiseRate* and *NoiseAmount* described in Section 5.8.4.

Next, the design of the fuzzy outputs of the model of audio expertise is described.

5.8.1 Output - Cluster

The sound design process involves clustering the harmonics into perceptually similar groups that the audio expert would choose using his experience in modeling pipe organ sounds. The fuzzy system has 10 different clusters from which the harmonics should belong. Note that some clusters may end up empty after the clustering process. The *Clusters* fuzzy variable has been created using 10 different fuzzy sets as follows:

- **Cluster10** corresponds to a cluster of harmonics that are grouped to model the general attack of the sound,
- **Cluster11** corresponds to a cluster of harmonics that are also grouped to model the attack segment if they are considered to be too different to be from *cluster10*. This cluster will deal for example with harmonics that have a different evolution from the rest with common attack and having interesting decaying segment for example.
- **Cluster20** corresponds to a cluster of harmonics that are grouped to model the generic sustain waveform. The term *generic* is used here in the context of *most common sustain waveform* and which differs from the following to *Cluster21* and *Cluster22*. The borders of belonging to these clusters are defined from the fuzzy sets of the input variables.
- **Cluster21** corresponds to a cluster of harmonics that are grouped to model the sustain waveform which has particular amplitude modulations. This cluster uses mostly the modeling rules dealing with amplitude modulations.
- **Cluster22** corresponds to a cluster of harmonics that are grouped to model the sustain waveform which has particular frequency variations. This cluster uses the rules about the frequency variations. Amplitude and frequency variations have different perceptual effects and influence the final sound quality in different manner.

- **Cluster30** is a cluster used to model the noisy waveform that is added to the overall waveform,

- **Cluster31** is a cluster used to model the noisiness that is to be added to the amplitude characteristics adjusted by two parameters as the amount and rate or variation (related to number of period of Musicom ADE time constant).

- **Cluster32** is a cluster used to model the noisy signal added to the frequency characteristics, also adjusted in both amount and rate.

- **Cluster40** usually for harmonics that do not fit into any of the previous described clusters. This cluster has been created to isolate particular cases the development of the fuzzy model would encounter. When a harmonic is assigned to this cluster, a process of deeper analysis is requested to the audio experts for deeper analysis and special care. Using this cluster additional rules have been created and added to the fuzzy model, as well as fuzzy variables characteristics adjusted (i.e. changing the fuzzy sets).

- **Cluster50** is a cluster added to cope with very high frequency. When sounds have high background noise, this indicate. This cluster will be used for future work when dealing with the sound pre-processing and particular aspect of pipe organ noise reduction. More work is needed in noise reduction for pipe organ sounds and it is the author beliefs that Fuzzy Logic techniques can help greatly.

5.8.2 Output - Attack Sound Synthesis

The attack sound synthesis fuzzy outputs are used to generate control parameters of the sound synthesis engine. All the fuzzy sets used for these fuzzy outputs are shown in Figure 5.10(a) to Figure 5.10(f). The Amplitude Envelope fuzzy sets are shown in Figure 5.11(a), whereas Figure 5.11(b) shows the ones used for the Frequency Envelope fuzzy

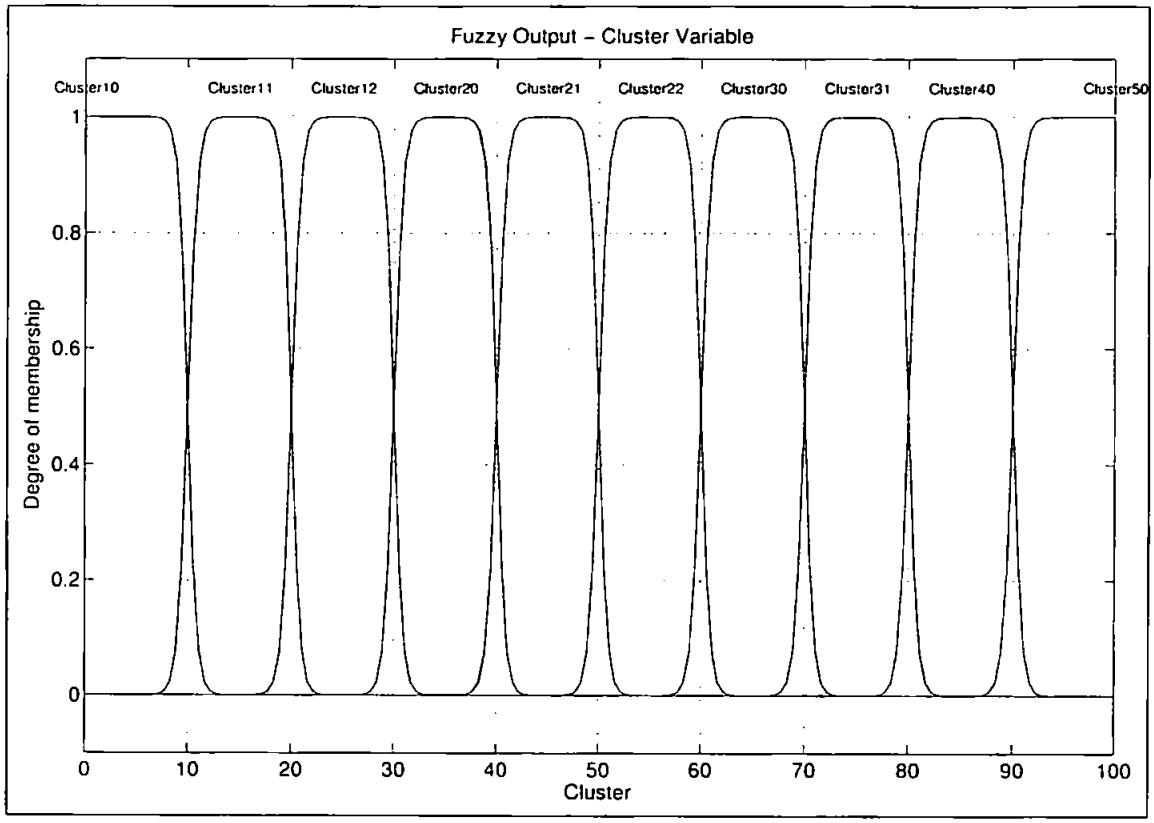


Figure 5.9: Fuzzy sets - Output Cluster

variable. The Amount of Amplitude Modulations fuzzy sets are shown in Figure 5.10(c) and the Rate of Amplitude Modulations in Figure 5.10(d), the Amount of Frequency Modulations fuzzy sets are shown in Figure 5.10(e) and finally, the Rate of Frequency Modulations are shown in Figure 5.10(f).

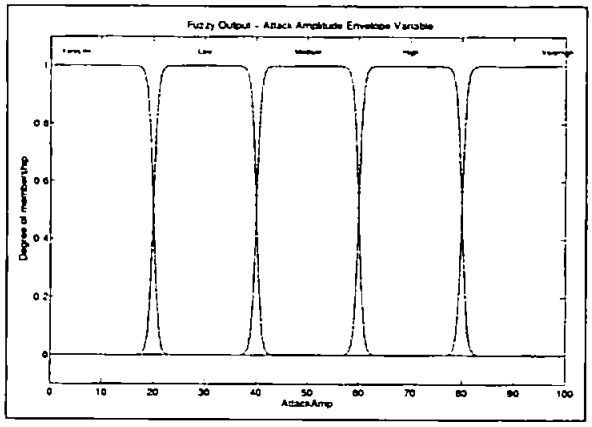
5.8.3 Output - Sustain Sound Synthesis

The sustain sound synthesis fuzzy variables are similar (in their name) to the one used for the attack, but applied to the sustain sound synthesis generator. Note that in Musicom terms, attack and sustain sound synthesis generators are the same (in terms of hardware resources), one being delayed to sound after the other, i.e. the sustain segment starting at the end of the attack time (See Section 4.3.1).

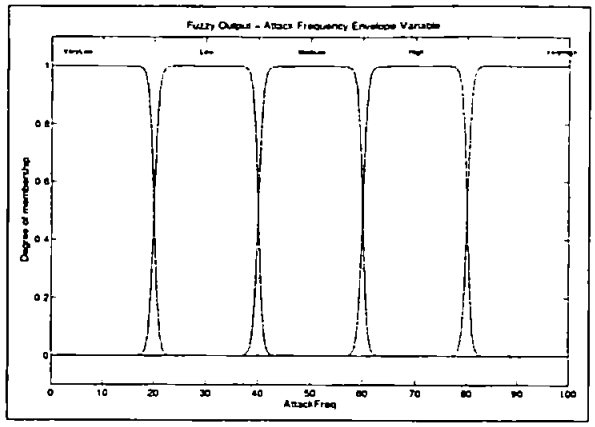
These sustain sound synthesis variables have been designed very closely to the Musi-

com ADE system, result from many years of developments by the audio experts in high quality sound synthesis. The audio expertise used here is really from both the audio experts themselves and expertise they have put into the Musicom ADE system.

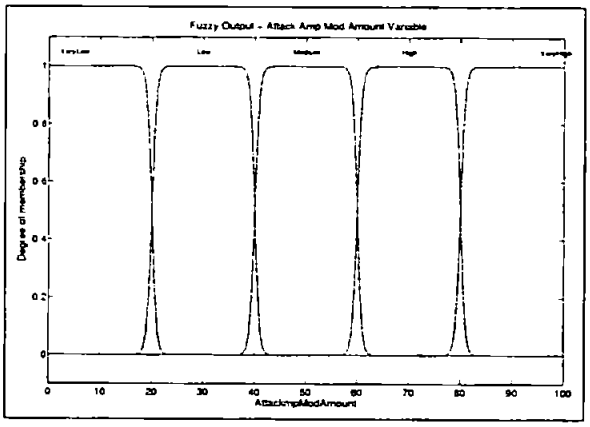
All the fuzzy sets for the sustain sound synthesis variables are shown from Figure 5.11(a) to Figure 5.11(f). The Amount of Amplitude Modulations fuzzy sets are shown in Figure 5.11(c) and the Rate of Amplitude Modulations in Figure 5.11(d) whereas the Amount of Frequency Modulations fuzzy sets are shown in Figure 5.11(e) and the Rate of Frequency Modulations in Figure 5.11(f). These are modulation adding realism to the sound on the top of the amplitude and frequency envelopes defined by the Amplitude Envelope fuzzy sets shown in Figure 5.11(a) and Figure 5.11(b) for the Frequency Envelope.



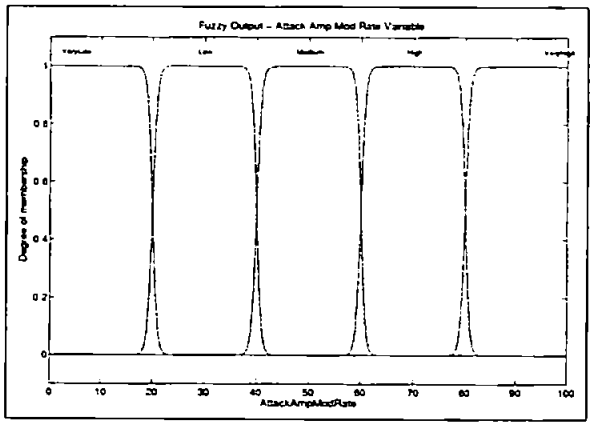
(a) Amplitude Envelope



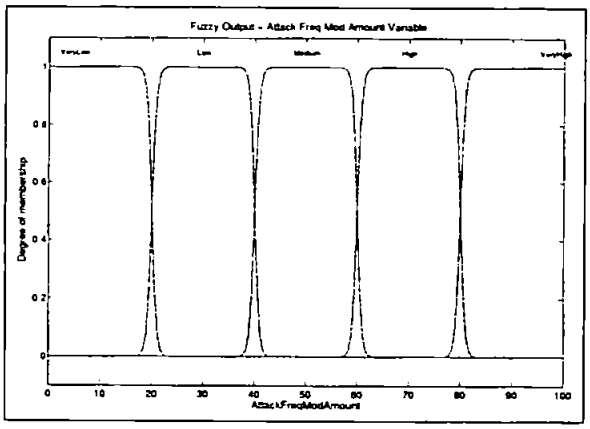
(b) Frequency Envelope



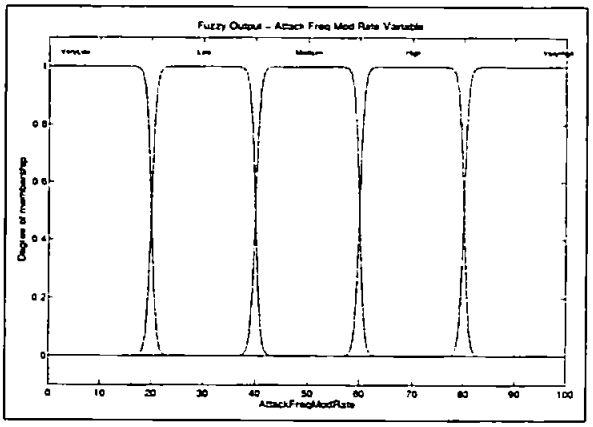
(c) Amplitude Modulation Amount



(d) Amplitude Modulation Rate

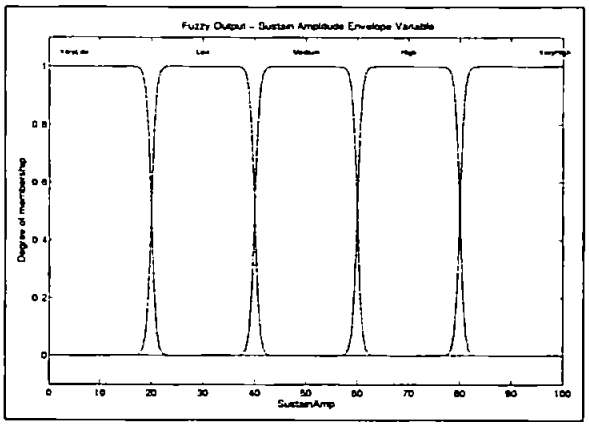


(e) Frequency Modulation Amount

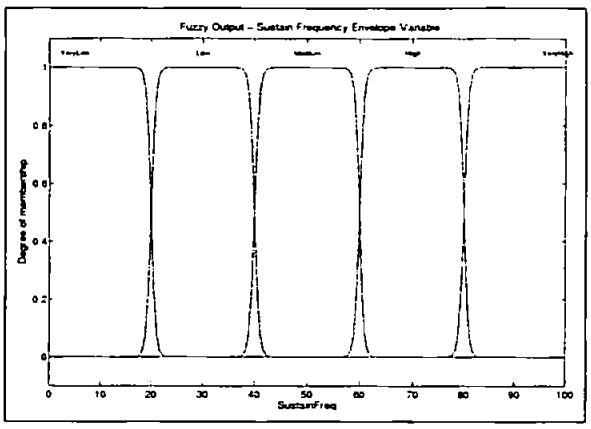


(f) Frequency Modulation Rate

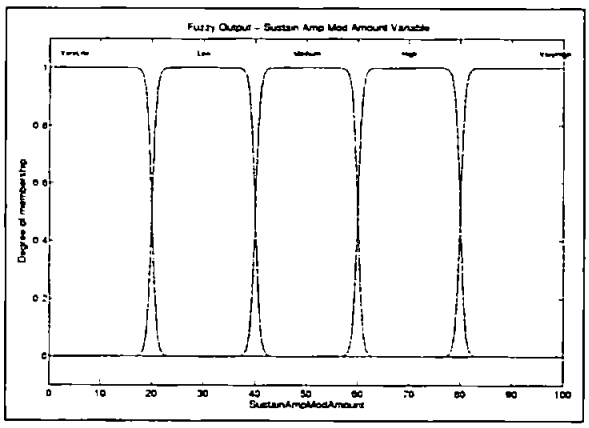
Figure 5.10: Fuzzy sets - Outputs Attack



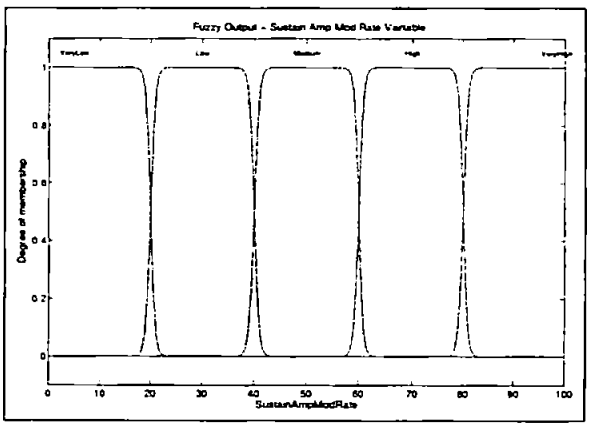
(a) Amplitude Envelope



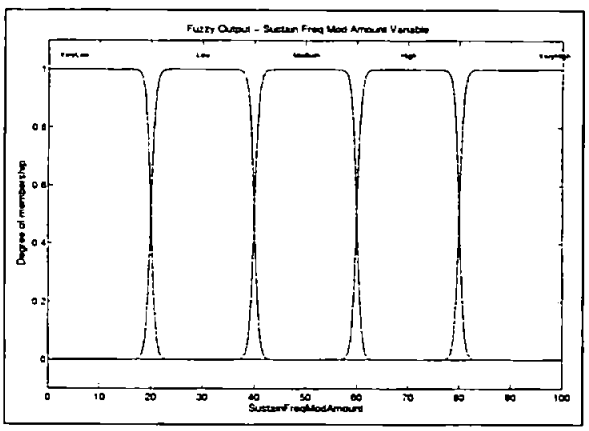
(b) Frequency Envelope



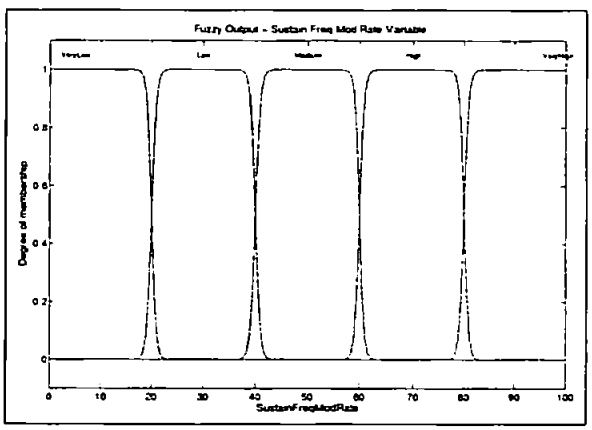
(c) Amplitude Modulation Amount



(d) Amplitude Modulation Rate

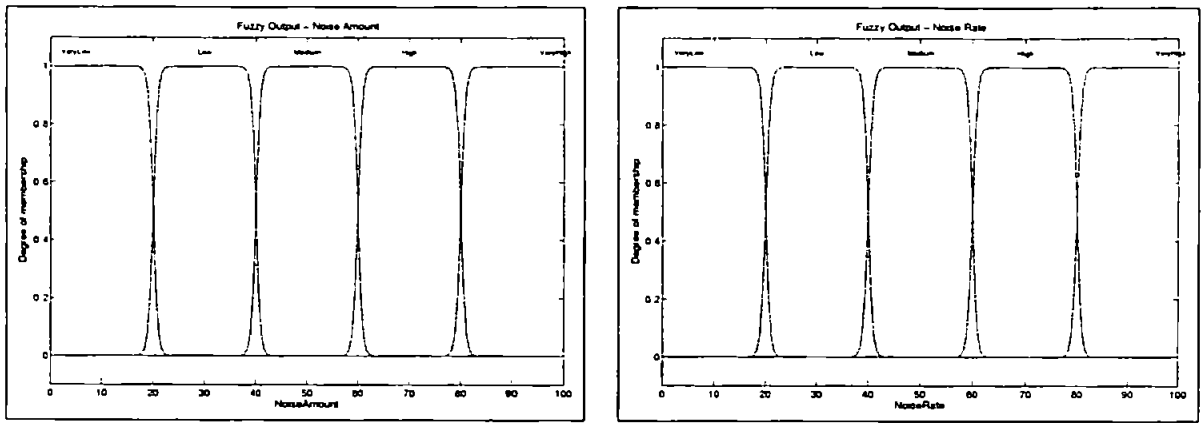


(e) Frequency Modulation Amount



(f) Frequency Modulation Rate

Figure 5.11: Fuzzy sets - Output Sustain



(a) Noise - Amount (b) Noise - Rate

Figure 5.12: Fuzzy sets - Output Noise

5.8.4 Output - Noise Generator

The noise generator is the final signal added to both attack and sustain generators to obtain the final sound. This noise generator is controlled by two parameters called NoiseAmount and NoiseRate which have corresponding fuzzy variables NoiseAmount and NoiseRate. Each variable consists of 5 fuzzy sets named *VeryLow*, *Low*, *Medium*, *High*, and *VeryHigh*. The maximal values corresponding to 100 is the latest Musicom ADE parameters from the VASM user manual [99]. These fuzzy sets are shown in Figure 5.12(a) and Figure 5.12(b) for the NoiseAmount and NoiseRate respectively.

5.8.5 Discussion

The variables and sets of the fuzzy model have been implemented with MATLAB scripts in which the their definition are parameters (defined as MATLAB vectors) and thus, easily integrated in the optimisation loop described in Section 3.6. The fuzzy variables and sets can be tuned and optimised using optimisation techniques (and tools like the Matlab Optimization Toolbox [20]) and create for example a pipe-specific fuzzy model of audio expertise (See Section 5.9.7), using sounds from the database of only Diapasons or Flute for example.

The implementation of the variables and sets is described in more details in Section 5.10 with the Appendix B.3 and B.4 for the fuzzy inputs and Appendix B.5 and B.6 for the fuzzy outputs. The definition and design of the fuzzy sets and have been is inspired from the Musicom ADE. However it should be understood that the methodology, presented in this thesis, is generic and can be easily adapted for other types of sound synthesis and as well as other musical instruments. Frequency Modulation synthesis for example, with its reduced set of parameters ($w(t)$ the carrier amplitude envelope, f_c the carrier frequency, f_m the modulator frequency, and I the modulation index, see Section 2.4.4) is an ideal candidate to use this approach and investigate the modeling of church bells. A vast amount of knowledge on FM synthesis is already available from the literature [72]. This will be carried out in future work.

The next sections present the fuzzy rules, definition of the audio knowledge used by the audio experts to create synthesis parameters from the audio features.

5.9 Fuzzy Rules from Audio Expertise

The fuzzy rules, described in the following sections, are based on the knowledge gained during elicitation sessions as well as results from the research project progression and feedback from the audio experts. In many respects, the fuzzy rules represented here are only a reduced subset of the audio experts' vast knowledge, and yet the most important rules used when modeling pipe organs and during the sound design process. Developments of the fuzzy model of audio expertise presented in this thesis are a first time in sound synthesis. It will become evident to the reader that some of the rules have links to known psychoacoustic facts [180].

Accurate modeling of musical instruments requires multi-level audio knowledge. The fuzzy rules, described in the following sections, are dealing with timing, amplitude, frequency, amplitude modulation, frequency modulation, and noise issues. They constitute

modeling knowledge used by the audio expert to design sounds. Other rules are also described, that apply to a higher level than sound design, and represent audio knowledge being "stops/pipes specific". Finally two examples are given, of rules applied to the highest level, the instrument itself, one on a particular aspect of a pipe organs called the "wind demand".

The fuzzy model uses the *centroid* as method of defuzzification, *min* as implication method, and finally *max* as aggregation method. It is not claimed to be best method for the fuzzy model, and can be changed easily from within the MATLAB scripts, as one parameter of the fuzzy model. Evaluation by the audio experts will provide clues about how to change these to improve the fuzzy system performance. The current fuzzy model should be considered as a prototype, illustrating the feasibility study of the novel concept of the Artificial Intelligence approach to sound synthesis parameters optimisation using audio expertise. The fuzzy model is continuously improved with the help of the audio experts.

5.9.1 Rule - Timing

The first rules of the fuzzy model deal with the timing, related to the Attack, Decay and release audio feature described in Section 4.3.1. The role of these rules is to detect the fast raising harmonics and cluster them accordingly.

First, the harmonics that have *Fast* or *VeryFast* attack characterisitcs are isolated as the represent the harmonics that influence the most the attack portion of the sound and attributed to Cluster10. These rules are also important to distinguish group of harmonics used for the attack and others of the sustain group.

Furthermore, the Timing fuzzy rules can also detect special cases like the harmonics that have an additional Decay audio feature and flag them with Cluster11. From the audio expert experience, harmonics which overshoot happen for Diapasons sounds only, and are usually present at the second harmonic. Using the Harmonic fuzzy variable, this

specificity of Diapasons can be dealt with. In future, further discussion with Musicom organ builders collaborators will help clarify the acoustics attributes of Diapason sounds to improve these rules and develop Diapason specific rules as described in Section 5.9.7.

5.9.2 Rule - Amplitude

When harmonics have predominant level, they tend to be responsible for the main sound character. Fuzzy rules Amplitude have been defined to group harmonics which Amplitude fuzzy variable falls into similar fuzzy sets, with priority given to harmonics with *VeryHigh*, *High* and *MedHigh* amplitudes (in that order).

Amplitudes that are *VeryHigh* or *High*, *MedHigh* or *Medium*, *MedLow* or *Low* can also be grouped together, which provides another sub-level grouping. Amplitude rules define the harmonic's contribution to the cluster, which in case of Musicom ADE would constitute the waveform used by one synthesis generator.

Using a combination of the Amplitude and Harmonic fuzzy variable allows to group harmonics that, with the perceptual masking results of the sound analysis engine as described in Section 3.3.3, could be potentially masked and thus reduce the synthesis resources used by the sound synthesis engine.

Harmonics with *VerySmall* Amplitude can be considered as noise and would fall in Cluster40 or even Cluster50. Experiments described in Sections 3.3.2 helped to further understand these influences and refine the Amplitude rules. For sounds with very complex harmonic structure, more fuzzy sets and rules can be added to the fuzzy model to improve the whole clustering process.

5.9.3 Rule - Frequency

Fuzzy rules Frequency have been defined to adjust the Frequency fuzzy variable of the harmonics fallings into similar fuzzy sets, with priority given to those within the *Low* and

Medium.

The fuzzy rules dealing with frequency aspects of sound synthesis do not participate directly in the clustering like the ones with dealing with the amplitude. The audio experts would be looking closer to the instability of frequency fuzzy variables (i.e. modulations) than the frequency fuzzy variables themselves.

In the classical voicing process of pipes, some pipes can be made slightly out of tune for sound characteristic purposes. This issue can be also dealt with using these Frequency fuzzy rules. Furthermore, the Brightness fuzzy variable can be used to adjust some the frequency fuzzy output of the sustain sound synthesis parameters in ways that help to render the characteristic of the sound into a more pleasing sound (enhancements like "exciter" are often adjusting the top frequency). The rules relating to frequency are difficult to elaborate (as they are often very "subjective").

5.9.4 Rule - Amplitude Modulation

Each harmonic has amplitude modulation, variations of the amplitude in the sustain part of the sound that gives the sound its character. Rules have been defined with the audio experts to try to obtain a good image of these variations. Harmonics with *Low*, amplitude modulation and *Medium* are of interest for these rules.

Cluster21 is used to indicate how much the harmonic is part of the sustain modulated cluster. The rules used for the amplitude modulation tries to provide the good amount of amplitude variation and at the right rate of change to obtain a clear sustain portion of the sounds similar as much as possible to the original sound. The VASM manual [99] provided insights about the Musicom ADE system with its parameters setting. This was define with the help of the audio experts.

There are some overlapping between rules for amplitude modulation and noise, for which the amplitude modulation can be considered as very small. This is defined by some of the Musicom ADE system.

Sounds with harmonics having *VeryLarge* amplitude modulation are considered suspicious and are assigned to Cluster50 (often cases of hum from bad recording conditions).

5.9.5 Rule - Frequency Modulation

It was noticed by the audio experts, that frequency instabilities of certain harmonics are often more noticeable than amplitude modulations. This fact is an important issue in this work as most current literature put emphasis is on the amplitude characteristics (envelopes and modulations) and often leaves very little consideration to the frequency aspect in sound modeling. The Fuzzy rules dealing with frequency modulation have been elaborated to isolate the harmonics with distinct frequency variations (Cluster22 and Cluster32). This is often seen from the analysis engine results (See Figures 4.6) and Figures 4.7) and the fuzzy variables Frequency (See Sections 5.7.6 and 5.7.8). The fuzzy rules dealing with the frequency modulation relate to the fuzzy output Cluster and especially fuzzy set Cluster22.

5.9.6 Rule - Noise and Randomness

The noise aspect of sound synthesis is critical and a key issue for adding realism to the sound at the re-synthesis stage. Musicom audio experts have improved the original design of the Musicom ADE to accommodate many of these features with better control and characteristic of the randomisation of the sound synthesis parameters, considered the key aspect of high quality sound synthesis. Without this randomness, the pipe organ just sounds very *clinical* and artificial.

Following the description of the Musicom ADE sound model by the audio experts, the fuzzy variable Cluster has been added to fuzzy sets to deal with noise aspect in terms of amplitude and frequency with Cluster31 and Cluster32 respectively.

The Noise and Randomness rules are trying to provide the fuzzy outputs of the model

(attack and sustain sound synthesis generators) with small amount of noisiness improving the perceptual impression of the sound. These rules are most difficult to define, and as they do not follow any psychoacoustic knowledge from the literature. This is the real experience of the audio experts with years of experience in sound synthesis. The rules of the current fuzzy model can be considered to be not as general as the other aspects of the fuzzy model. They are based on the pipe organ sounds and by their acoustic nature pipe organ sounds are often very breathy.

The current system is not able to make difference between the noise components of the sound and the noise from the recording conditions for example, and the results often depend on the quality of the original data. The system tries to model the original sound and has no clue if the sound conditions are bad or just part of its noisy breathy characteristic.

The audio experts referred the chiff as "the noisy burst is usually cause by the imperfect response of the pipe to the shock wave of the initial airflow start" (acoustics about jet formation is described in [162]) and is partly harmonic, and partly noise. The pitch of the harmonics are not the same as in the sustain tone. This chiff character is typical for different types of pipe, several basic types of pipe have their own characteristic chiff and sustain harmonics and include open and stopped flutes, principals (diapasons) and reeds.

5.9.7 Rule - Stops/Pipes specific

The fuzzy model can deal with audio expertise issues and helping the modeling of musical instruments at a higher level than the sound (one single key). The fuzzy model can incorporate modeling rules dealing with key range and influence of both attack and sustain on the sound synthesis outputs for specific type of stops/pipes (e.g. reeds, flutes, etc.). The refinement of the rules is an continuous collaboration with the audio experts [97].

Reeds - Reeds have, as their name suggests, a mechanical "exciter" that develops a much

stronger harmonic series, being usually more stable than other pipes during the sustain period. From the audio expert's experience, this can therefore be re-simulated much more easily, with usually few clusters across the entire range of notes. The attack portion can require an additional sound generator.

Diapasons or Principals - In the case of Diapasons or Principals, they have a strong low order harmonic structure, but typically have a group of harmonics around the 15^{th} to 25^{th} that are unstable. This instability is usually present in both amplitude and frequency and this can be related to the acoustics: as their wavelength approaches the width dimension of the pipe. Audio experts always insisted that instability is a key item in re-synthesising their characteristic. From the audio experts point of view, this particular aspect could be ascertained by studying the physical dimensions of such pipes. Diapasons or Principals can be constructed from wood (in which case they have square cross section) but mainly in metal (in which case they have a circular cross section) [71].

Flutes - Flutes are usually constructed in wood (with a square cross section) and are either "open" - open ended (full length 'resonator') or 'stopped' (half length resonator with a stopped end). Flutes usually have a strong fundamental and a limited number of stable harmonics, progressing rapidly into noise, as shown from the sound analysis with the harmonic distribution shown in Figure 4.3(b). One easy recognisable characteristic of stopped pipes is that they have predominantly Odd order harmonics. (See Figure 4.3(d)). Finally, they present some instability as the harmonic wavelength reached the width dimension of the pipe (as in the case of Principal).

Mixtures - Mixtures are usually very small pipes, and a combination of flutes and principals. This has not been really a consideration in this research work as the audio experts "construct" them using small pipe characteristics, considered much simpler to re-synthesise.

5.9.8 Rule - Pipe Organ

The current fuzzy model provides an attempt to define rules with the audio expert to design sounds. This can be considered as a low level aspect of modeling a musical instrument. The fuzzy model incorporate knowledge that apply to a higher level, i.e. at a musical instrument level. Two facts about the pipe organ are used to illustrate this idea, the first is that "complexity of pipe organ sound decreases with increase in pitch" and the second is the issue of "wind demand".

Complexity of pipe organ sound decreases with increase in pitch This fact often repeated by the audio experts is very important as it make the fuzzy model to priorities the modeling process to allocate more resources to the low pitched sounds than the high pitched one. The fuzzy variable Pitch can be used to check from which octave the sound belong to, and more advanced rules can make use of the Harmonic variable to assess the harmonic distribution like the audio expert does visually from Figure 4.3 and Figure 4.4. The two main reasons, given by the audio experts [97] and supported by psychoacoustic principles [180], come from that first the human hearing has a upper limit at 20 kHz (similar to a Nyquist[‡] type of limit with exceptions for "Golden ears"). Secondly, this is combined the fact that the number of harmonics available produced in the physical structure of the pipe conveniently follows that Nyquist limit, so as the higher the pitch, the lower the number of harmonics required. As an example if a note has a fundamental 2000 Hz it will only have "possible" 10 harmonics, whereas tones in the lower range, for example at 200 Hz the Phase Vocoder can extract 100 harmonics. This fact can be easily demonstrated with WinPVoc with which some sounds from the analysis generate a 365 harmonics (sampling frequency is 48 kHz and fundamental is around 65 Hz), some of which in the very high part are only noise.

Wind Demand. Another "high level" aspect of pipe organs relates to the wind demand.

[‡]as defined in [78] by the process of digital audio sampling

Wind demand is an important feature of the pipe organ instrument [71]. It can be simply explained as follow. When many keys are pressed simultaneously, the total wind pressure required to sustain the different sound of each pipe may exhibit some fluctuations, producing some slight drift in the pitch of each pipe's sound. The Pitch fuzzy variable of the fuzzy model can be used with rules that would render these fluctuations and thus greatly improve the sound of an electronic pipe organ. To accurately define these rules, measurements on the distribution of air pressure should be carried out on real instruments using multi channel recording (the wind distributing portions of pipe organs can be found in [71]) and use WinPVoc to analyse the pitch variation of the sound of each pipe, or use some ICA[†]-based techniques to separate them as sources [55]. In a system like the Musicom ADE, the keyboard manager unit (MIDI processor) can monitor the MIDI stream and the number of keys pressed to apply this *wind demand rule*.

5.9.9 Discussion

The fuzzy rules that define the audio expertise can be considered at different levels: at a sound level for each individual key, for a range of keys (whole octave for example), for individual stop/pipe, and finally at the highest level, the musical instrument itself as seen for "wind demand". The research project has been focused on aspects of sound design and these *higher level rules* have not been incorporated into the current fuzzy model. However, the complexity of the pipe organ comes from the additive contribution all each pipes and the way they all interact with each other in a piece of music. From the audio experts' experience as musicians and organists (Graham Blyth is known for his recitals at Audio Engineering Society conventions), in a musical performance all these small "details" become audible artefacts that the human ear detects very quickly.

The fuzzy model can also be used to help improve the process of "electronic voicing" (and functionalities of the Musicom PipeSpecBuilder tool [122]), in similar ways organ

[†]Independent Component Analysis [76]

builders are currently voicing individual pipes [133]. Voicing is a key aspect of pipe organs and a very complex and challenging task that requires skills and experience. Research work in this direction with the help of both audio experts and organ builders collaborating with Musicom Ltd is in preparation. All these issues are under investigations as part an EPSRC funded project [36] extending the present work described in this thesis.

Finally, it is important to emphasis that the definition of the fuzzy rules can be easily adapted for other types of musical instruments. Every musical instrument has its own characteristic and playing techniques. Piano for example have sympathetic resonance of high key strings. Woodwind players (flutes in particular) have a technique by in which the player directs the flow of air to obtain the pitch of the first overtone or harmonic rather than the fundamental pitch which would normally be sounded. All these demonstrate the importance of audio knowledge that should be incorporated into future development and design of electronic musical instruments with the help of audio experts.

5.10 Implementation of the Fuzzy Model

The fuzzy model presented in this chapter has been implemented using two software solutions: with MATLAB and its Fuzzy Logic Toolbox and with Microsoft Visual C++ 6.0 as a Fuzzy Library (C++/MFC) and FuzzyEditor. The next sections detail both solutions.

5.10.1 MATLAB and Fuzzy Logic Toolbox

The MATLAB environment with its Fuzzy Logic Toolbox has been found very convenient for the development of the fuzzy model. As described in Section 3.7, WinPVoc sound analysis engine extracts audio features and generates MATLAB scripts with variables used by the fuzzy model as parameters. The fuzzy model after processing these audio features generates sound synthesis parameters (MATLAB scripts with associated binary

data) that WinPvoc can import and use to re-synthesise the sound. In future, the fuzzy model will directly generate VASM files for the Musicom ADE system using the latest VASM language specifications [99].

The five editors of the MATLAB Fuzzy Logic Toolbox, as shown in Figure 5.13, can be used to create, edit and evaluate a Fuzzy Expert System (FES). A **FIS Editor** is the main editor calling 1) the **Membership Function Editor** used to create fuzzy sets with a variety of membership functions (sigmoid, gaussian, triangular, etc.) and 2) the **Rule Editor** to create and modify the fuzzy rules. Furthermore, the **Rule Viewer** helps to assess the effect of variables variations onto an output and the **Surface Viewer** allowing to see this effect as a 3D mesh surface. The MATLAB functions for these editors are *fuzzy*, *mfedit*, *ruleedit*, *ruleview*, and *surfview* respectively. More information about the Fuzzy Logic Toolbox can be found in [139].

The combination of these editors provides a comprehensive set of tools to support a successful development of the fuzzy model, and this from the design stage the the simulation and evaluation to of the fuzzy model. The toolbox supports customised rules generation, various type of membership functions, for creating standard Mamdani and Sugeno-type of Fuzzy Inference Systems (FIS).

Within the MATLAB environment the fuzzy model can also be easily integrated into the optimisation loop (Section 3.6) using simple Least-Squared (LS) optimisation techniques or more advanced techniques such as Simulated Annealing [43]. Note that the optimisation used in this research project is objective and perceptual as the Sound Quality Assessment Engine described in Section 3.5 is based on the PEAQ algorithm.

5.10.2 Fuzzy Library and Fuzzy Expert System

The long term of this research project is to provide the audio experts a fuzzy model to be integrated into WinPvoc, thus the fuzzy model has to be ported to C++. The MATLAB environment, with the digital signal processing and artificial intelligence toolboxes, is a

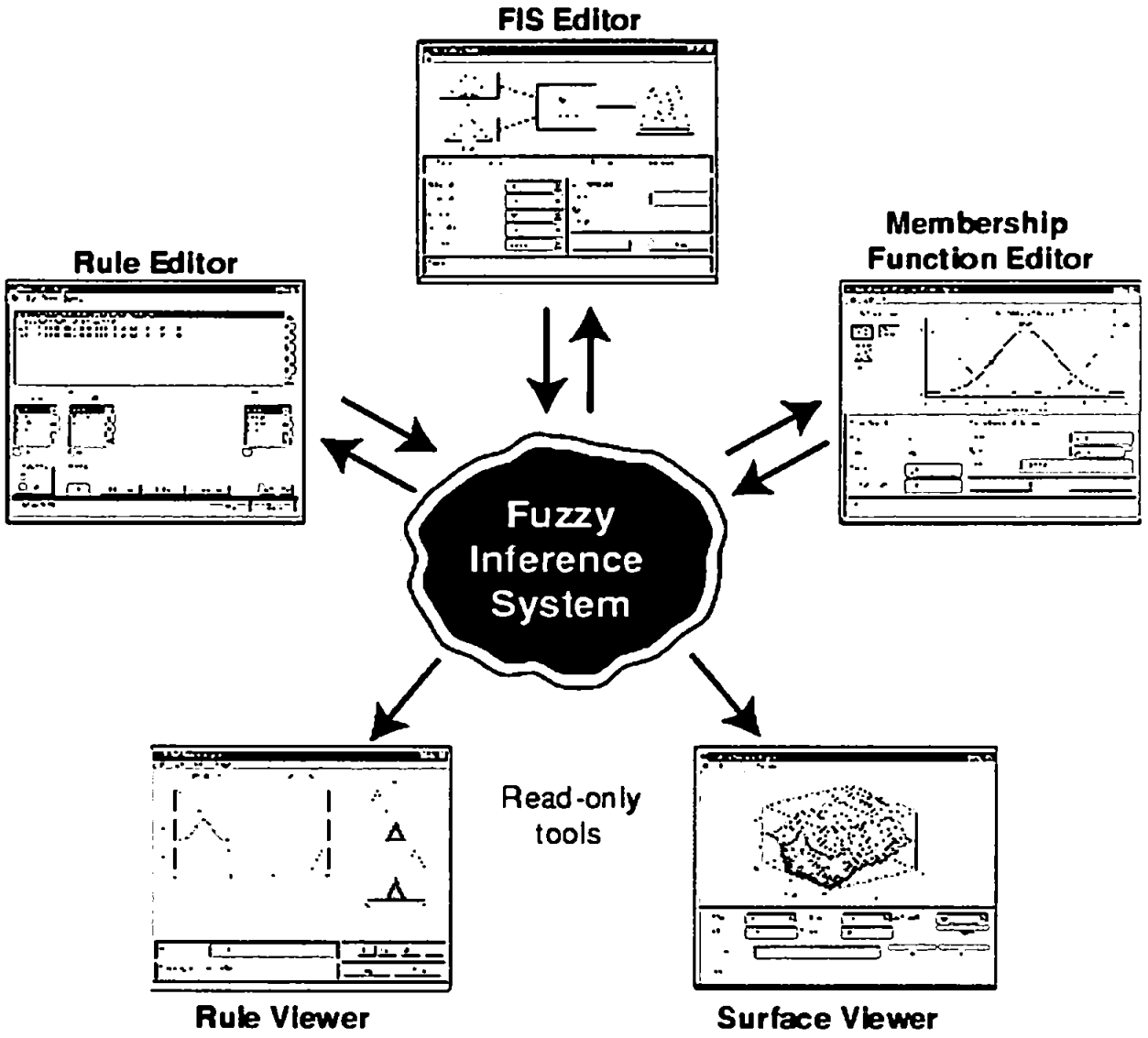


Figure 5.13: MATLAB Fuzzy Logic Toolbox

very popular tool for research and teaching purposes. However the time required to learn this new programming language and the licence costs were important issues for the audio experts and Musicom Ltd.

A fuzzy library has been developed in C++/MFC (Microsoft Foundation Classes) following guidelines from [25]. It consists of 4 main objects: CFuzzySet, CFuzzyVariable, CFuzzyRule and CFuzzySystem. The development of the fuzzy library aimed to provide most of the functionalities used from the MATLAB Fuzzy Logic Toolbox, with each

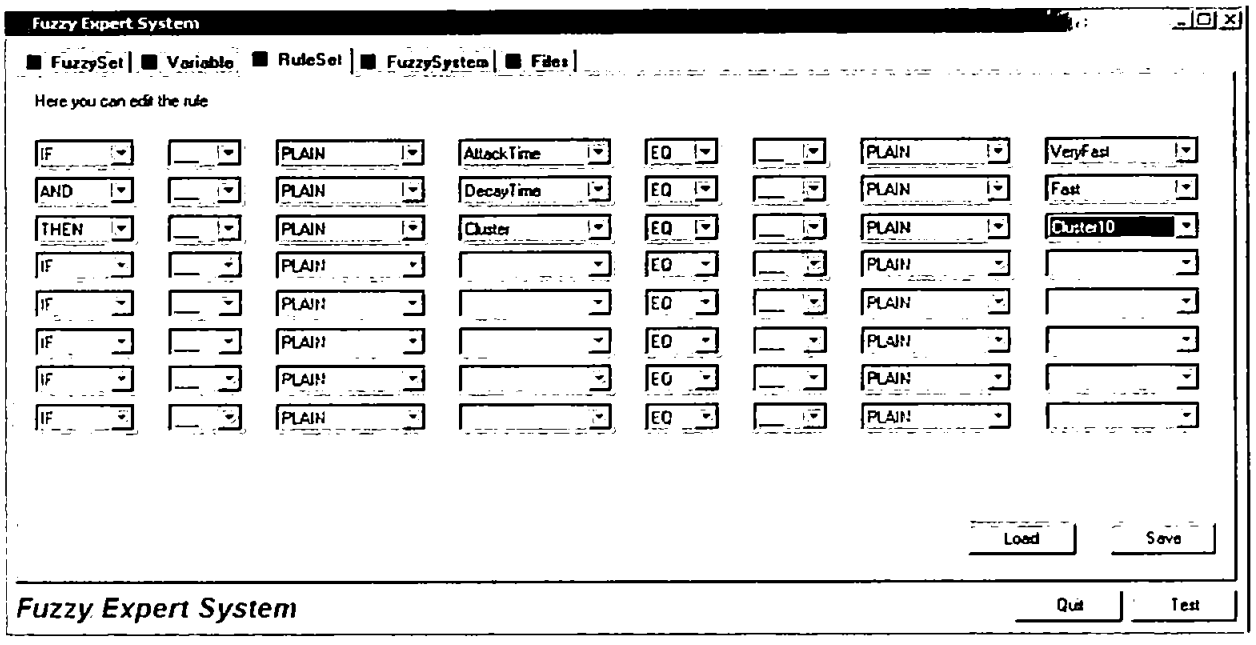


Figure 5.14: Fuzzy Expert System

object or class having its own member variables and methods to allow access to object characteristics. An associated GUI, in form of PropertyPages, has also been developed for each class.

A FuzzyEditor has been developed with five PropertyPages for 1) graphical edition of the fuzzy sets, 2) creation of fuzzy variable and addition of fuzzy sets, 3) creation of fuzzy rules (as shown in Figure 5.14), 4) creation of fuzzy systems with addition of fuzzy variables and rules, and finally 5) file management to save/load fuzzy sets, variables and rules from disk with options to import and export their definitions in Matlab scripts format.

The fuzzy library has been compiled (as an independent DLL[‡]) and linked in the latest version of the WinPVoc tool. Once the development of the fuzzy model of audio expertise is considered satisfactory by the audio experts, it can be used within WinPVoc, removing the need for the MATLAB environment. Finally, the fuzzy library and fuzzy expert system tool will be used in other research projects combining audio signal processing

[‡]Dynamically Linked Library

with Fuzzy Logic (See Section 3.2.1).

5.11 Summary

In this chapter, the development of the audio feature processing engine has been described. It is based on a fuzzy expert system implementing a fuzzy model of audio expertise designed in collaboration with two audio experts. This fuzzy model of audio expertise is main contribution of this research project. First, knowledge elicitation, the process by which the expert knowledge has been collected and formulated for the development of the fuzzy model has been described. The development of the fuzzy model with the elaboration of the fuzzy variables with their fuzzy sets, for the fuzzy inputs (audio features) and fuzzy outputs (sound synthesis parameters), as well as the fuzzy rules (decision making process), has been detailed. All have been defined in collaboration with two audio experts following on-going knowledge elicitation. The fuzzy sets and variables define, with linguistic terms that human can understand easily, the key audio features used at the sound analysis stage of pipe organs, while the fuzzy rules define the decision process used by the the audio experts for sound design. Finally, the fuzzy model has been implemented in MATLAB, with scripts to create the fuzzy sets, fuzzy variables, fuzzy rules, and fuzzy inference process. MATLAB scripts are given in Appendix B.3 and B.4 for the fuzzy inputs, Appendix B.5 and B.6 for the fuzzy outputs, and Appendix B.7 and B.8 for the fuzzy rules and fuzzy inference, respectively. A fuzzy library in C++/MFC[‡] has also been developed and integrated into the WinPvoc tool.

[‡]Microsoft Foundation Classes

Chapter 6

Evaluation and Performance

6.1 Introduction

This chapter presents the evaluation and performance of Intelligent Audio System, the novel Artificial Intelligence-based approach to sound synthesis parameter optimisation described in Chapter 3. The performance of the system is assessed from the perceptual sound quality of the synthetic sounds it produces.

The sound database is first described with the automated sound analysis. The methods used to assess the sound quality are presented with selected sounds and modeling results given for each of them. Finally, the significance of the results and benefits for the audio experts are discussed.

6.2 Sound Database

At the early stages of the research project, a set of four CDROMs with pipe organs recordings was provided by Musicom Ltd. These recordings are the result of recording sessions of pipe organs in Europe (England, France and Germany) and overseas (Canada, and mostly United States) and form the sound database.

Recordings are from organs including 1stCongregational (220 MBytes), Casavant (30 MBytes), EpiscopalHouston (126 MBytes), German (12 MBytes), Hexham (17 MBytes),

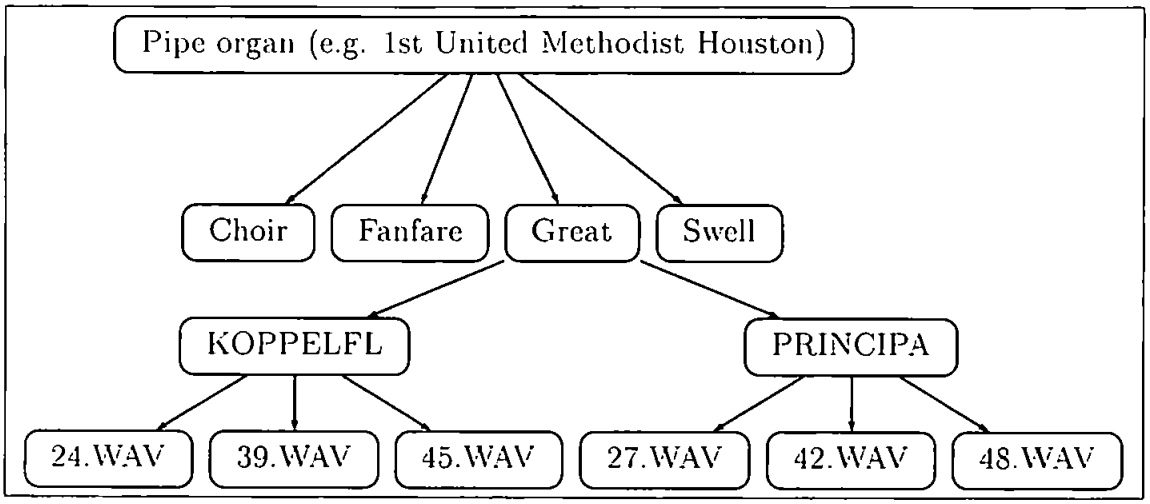


Figure 6.1: Tree diagram for pipe organ sound database

Houston (270 MBytes), HuntsvilleAlabama (189 MBytes), Huron (38 MBytes), JohnDivineHouston (119 MBytes), KilgoreTexas (111 MBytes), Knoxville (116 MBytes), SacredHeart (114 MBytes), Torrington (260 MBytes), Wicks (258 MBytes), in total representing 1.84 GBytes of sound data.

The sound analysis of the database is very time consuming. Two functions MATLAB scripts called CatalogueCDROM and CatalogueOrgan have been developed and used to automatically scan the sound bank structure and generate reports like the ones given in Appendices A.1. The function CatalogueCDROM scans the database (from an original path) for pipe organ folders, calling the function CatalogueOrgan that looks for registers (Great, Choir, Swell, etc.), and further down for pipe/stops (names and abbreviations of pipes) and finally for the actual the WAVE files. These functions generate formatted reports in LaTeX language [50] with tree diagrams like the one shown in Figure 6.1. More information about the sound analysis of the sound database can be found in [57].

Note that in Figure 6.1, *KOPPELFL* is an abbreviation of Koppelflute and *PRINCIPA* is for a Principal, wave files (24.WAV, 39.WAV, 45.WAV, etc.) are numbered following Musicom specifications shown in Table A.1.

The majority of the database sounds were in 16-bit PCM format in WAVE stereo

format, sampled at 44.1 kHz and following the Musicom format: one channel is based on the recording at the *mouth of the pipe*, while the second channel is at an specific distance or the pipe. This technique is often used by the audio experts. All the sounds were converted to 48 kHz using a high quality sample rate conversion feature of a sound editor[‡] (See Section 3.2.1).

6.3 Assessment of Sound Quality

The task of audio quality assessment has been carried out using methods described in Section 2.5. Working with audio professionals during this research project has made the final evaluation of sound quality to become very important. The methods used to assess the sound quality are the following:

- **Listening tests** - Listening tests have been organised at Musicom Ltd facilities to assess the audio quality of the sounds produced by the intelligent audio system. The audio experts can with the WinPVoc SDG Dialogbox send results via emails (See Figure 3.17). These listening tests were not strictly following the standard procedures specified in the ITU-R BS1116 document [80] which brief description was given in section 2.5.1. However, the audio experts have extensive experience in listening tests, and thus in the context of this research project, they are the best candidates for assessing audio quality of pipe organ sounds.

- **MSE / PSNR** - As described in Section 2.5.2, the "Mean Square Error" (MSE) (and associated "Power of Signal-to-Noise Ratio" (PSNR)) has been used traditionally used as error measure [169], even if recognised by many to poorly correlate with real perceived audio quality [18]. They are here used as the basis for comparison with other methods.

[‡] Wavelab 4.0c from Steinberg GmbH

- **WaveDiff** - The relative difference between two successive (note: successive number of clusters) waveforms has also been used. This measure will indicate any small distortion at a waveform level and also situate this error along the time axis (i.e. attack, decay, sustain and release portions of the sounds).

- **PEAQ evaluation** - The latest ITU-R standard in the form of the Perceptual Evaluation of Audio Quality (PEAQ) algorithm (described in Section 2.5.3) has been used to assess the audio quality of the sounds produced by the intelligent audio system. Initial experiments were carried out with a free version PEAQ algorithm called WinEaQual[†] implementing the basic version of PEAQ only, and modified to generate MOVs and DI/ODG results as MATLAB scripts (selectable from the Preferences of WinPVoc see Figure 3.16. Recent experiments are using an OPERA^{TM} Voice/Audio Quality Analyzer (pictured in Figure 2.8).

6.4 Results - Modeling by clustering

In the following sections present the results of the intelligent audio system used for the sound design process. More results together with the sound analysis of the database are being summarized in [57].

6.4.1 Choice of Sound Materials

The sound database A.1 provides a very large number of pipe organ sounds that can be used to evaluate the performance of the novel AI-based approach to sound synthesis parameter optimisation presented in this thesis. In the next sections, seven sounds have been used to evaluate the intelligent system and modeling results for each sound are presented. Pipe organs can produce a very large variety of musical tones and the selection

[†]Developed by Alexander Lerch `http://www.zplane.de`

of these sounds only followed the advice of the audio experts. These sounds have very different tonal characteristics and are considered as challenging example for modeling.

These sound are from the following stops: a Basson 8' sound with fundamental of about 108 Hz, a Flute sound with fundamental of about 527 Hz, a Horn sound with fundamental of about 110 Hz, a Principal 4' sound with fundamental of about 260 Hz, a Dulciana 16' with fundamental of about 131 Hz, a sound from a Metal Trombone 16' with fundamental of about 65 Hz and finally an Oboe 8' sound with a fundamental of 197 Hz. The Dulciana 16', Metal Trombone 16' and Oboe 8' sound were recently recorded by the audio experts from a Schantz pipe organ [144].

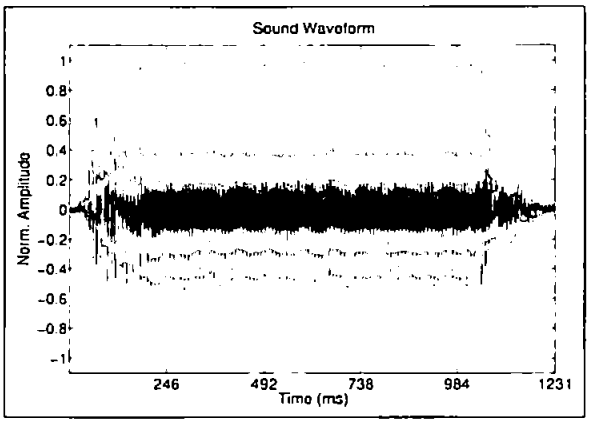
6.4.2 Sound - Basson 8'

Analysis - A Basson sound is use as the first example to present results. The analysis tool evaluated the fundamental to be 108.35 Hz. Figure 6.2(a) show the time domain waveform giving some indications about the overall envelope. Figure 6.2(b) shows the spectrum of the same sound, with clear harmonics and also some indications about the the noise floor level (high frequency part). To have a better view of the evolution of the harmonics, Figure 6.2(c) and Figure 6.2(d) plot them as harmonics lines in linear and logarithmic scales. The audio experts often prefer to work with the log scale as it brings out the noise floor. Figure 6.2(e) show the harmonics superimposed onto one graph with the sustain level (dashed line). It is also interesting to see the maximum and sustain level for all the harmonics as presented in Figure 6.2(f).

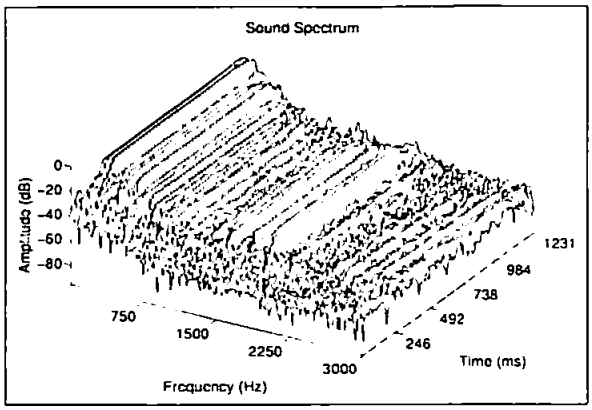
Dendrogram - Figures 6.3(a) to 6.4(l) and Figures 6.4(a) to 6.4(l) show results from the clustering algorithm with the different methods for calculating the metric distance and linkage methods. These figure show very different characteristics which are difficult to make conclusions from. Only the final sound re-synthesised can tell about their "goodness". Still the Dendrograms have different shapes which are recognisable as *spread* like in Figure 6.3(j), 6.4(c) and 6.4(k), or *rising* like in Figure 6.3(d) and 6.4(g).

Modeling Error - The clustering error results, using 4 different distance metric methods (Cityblock, Euclidean, Max and UQI) are shown in Figure 6.5, Figure 6.7, Figure 6.9 and Figure 6.11. With a small number of clusters, the case of Blk method is typical whereas the UQI method presents higher error (nearly to 1.0) but with a different shape (non-uniform). The Max method gives some error at the beginning of the clustering process, which indicates some wrong clustering, making the clustering error to rise too quickly. The Max method has been found unsuitable for most of the clustering experiments.

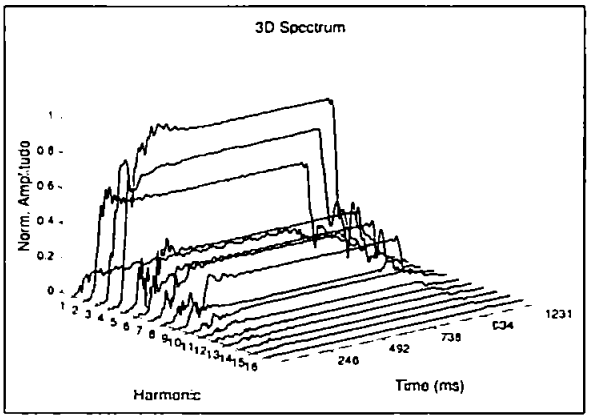
Objective Audio Quality - Figure 6.6, 6.8, 6.10 and 6.12 show the Objective Audio Quality resulting in the AHC algorithm with 4 different distance metric methods (Cityblock, Euclidean, Max and UQI). The Cityblock and Euclidean present similar results. UQI seems to provide slight improvements, looking at the intersection between the perceptible plan and the objective audio quality surface.



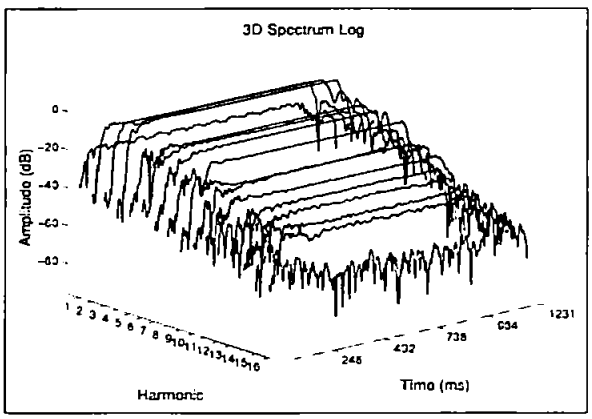
(a) Waveform



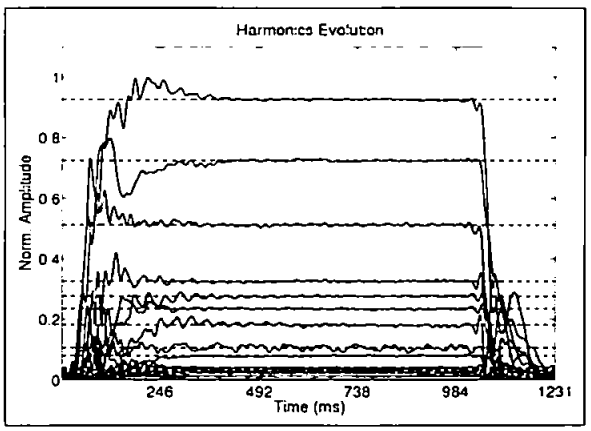
(b) Spectrum



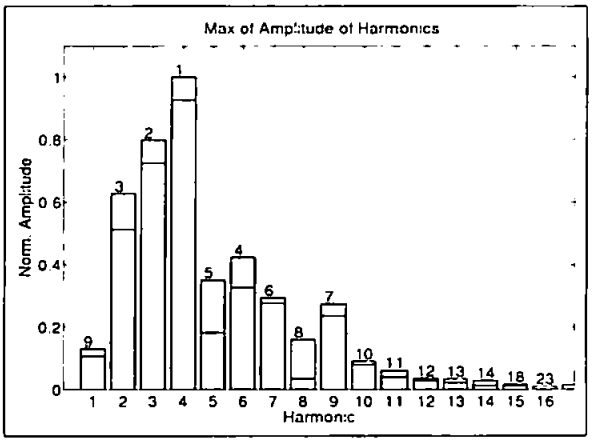
(c) Spectrum lines (Lin)



(d) Spectrum lines (Log)

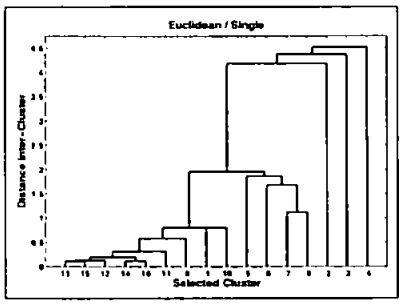


(e) Harmonics envelopes

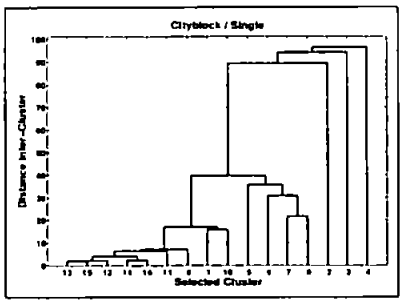


(f) Harmonics distribution

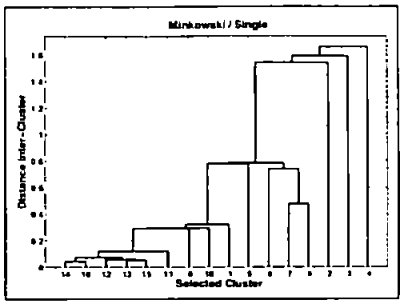
Figure 6.2: Basson - Sound analysis



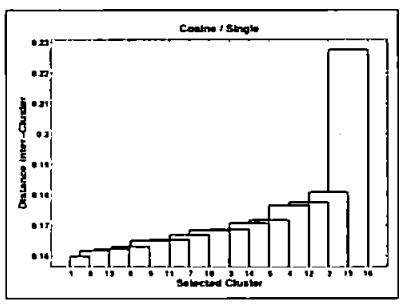
(a) Euclidean / Single



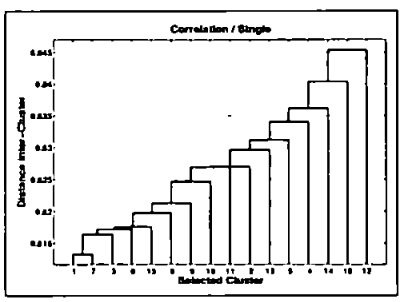
(b) Cityblock / Single



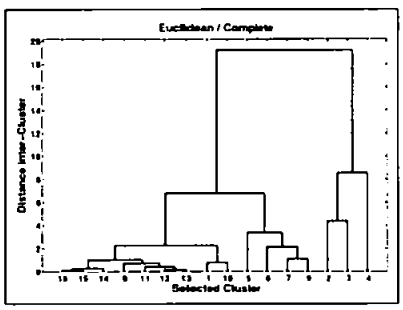
(c) Minkowski / Single



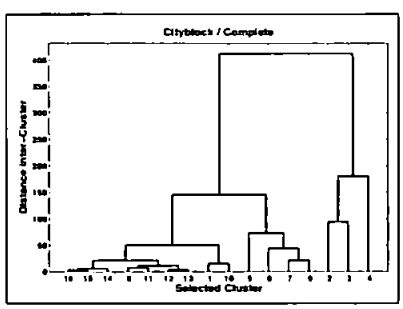
(d) Cosine / Single



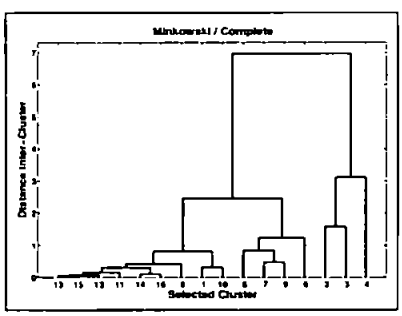
(e) Correlation / Single



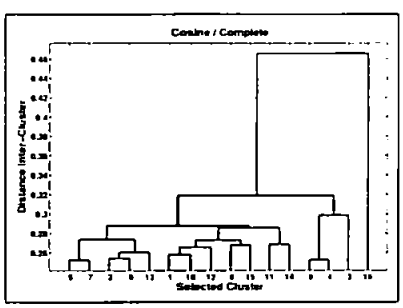
(f) Euclidean / Complete



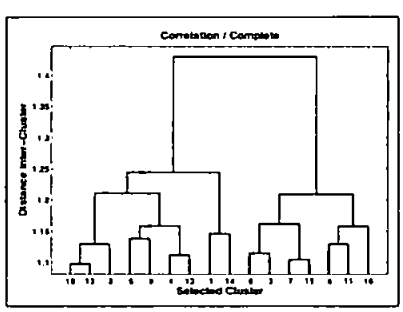
(g) Cityblock / Complete



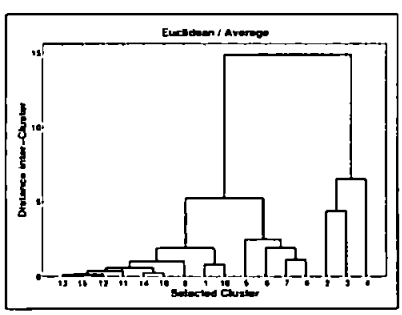
(h) Minkowski / Complete



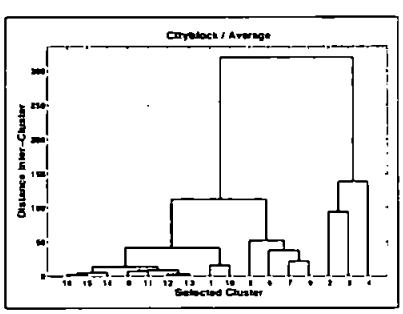
(i) Cosine / Complete



(j) Correlation / Complete

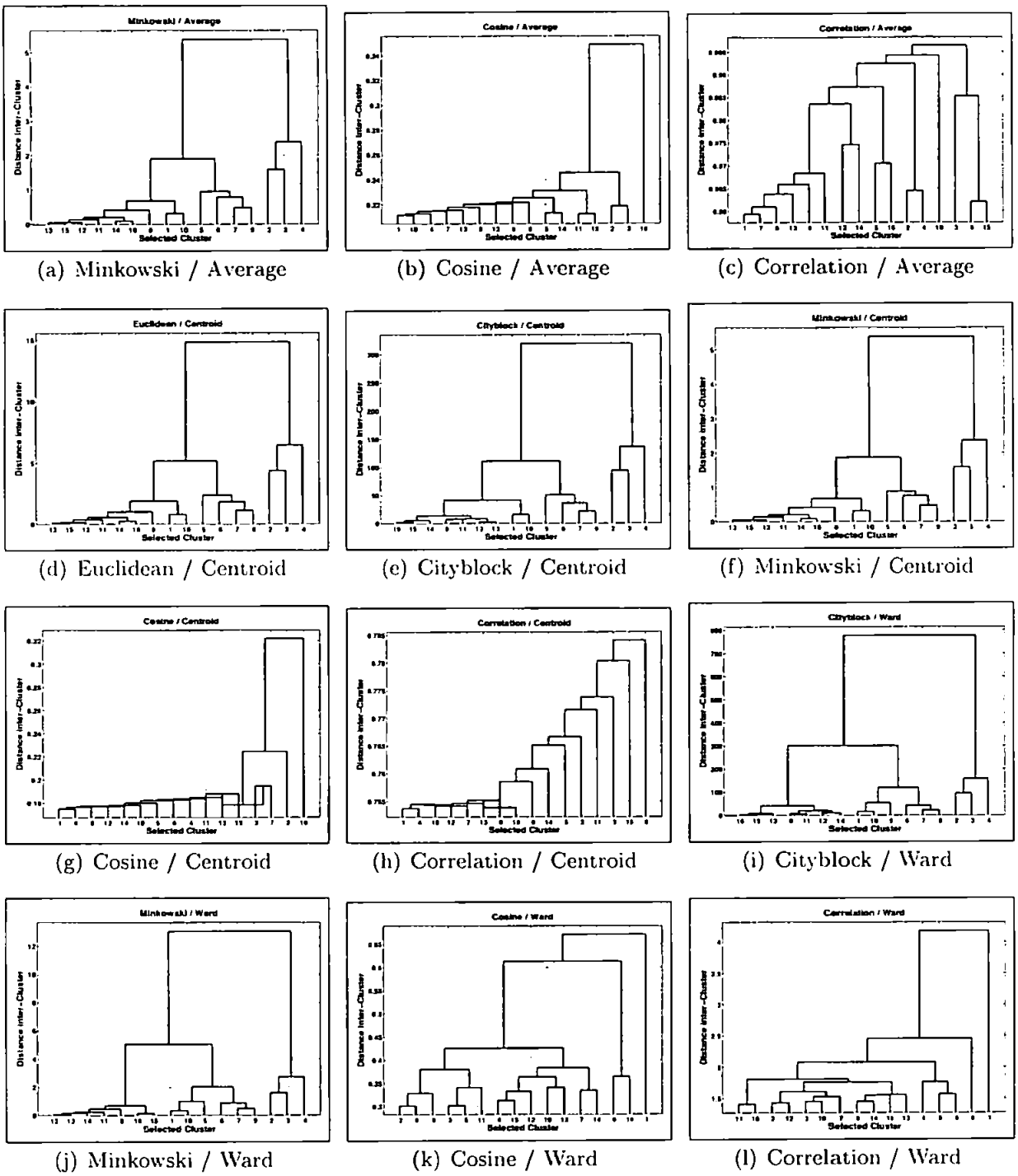


(k) Euclidean / Average



(l) Cityblock / Average

Figure 6.3: Basson - Dendrograms results



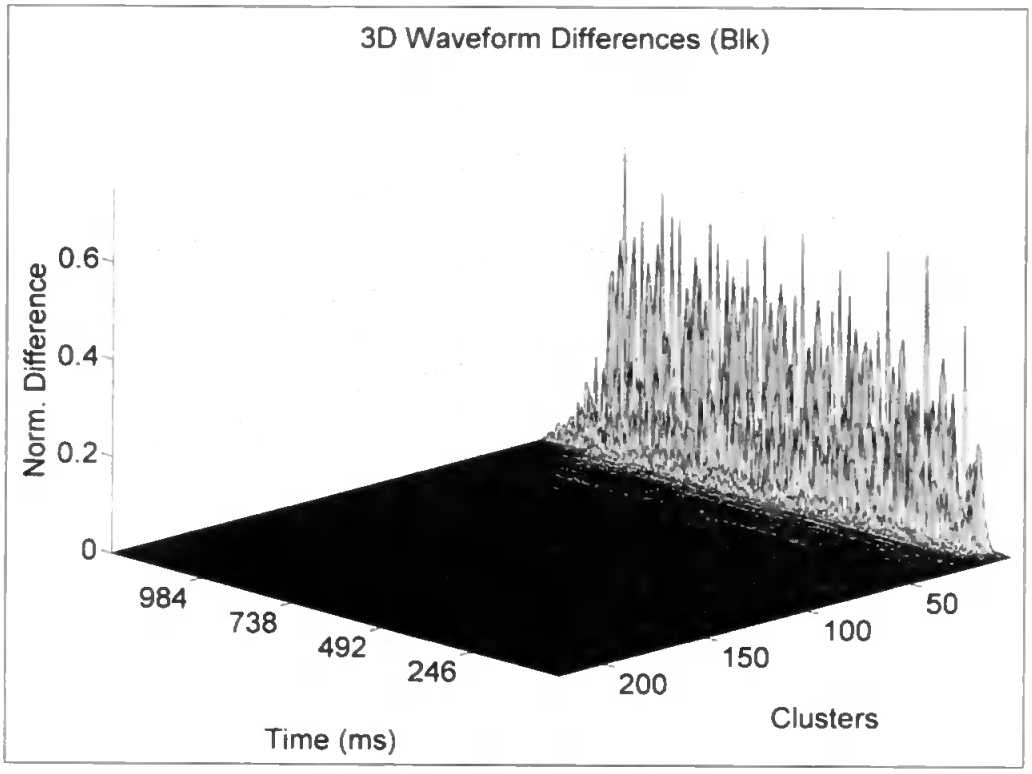
(a) Minkowski / Average (b) Cosine / Average (c) Correlation / Average

(d) Euclidean / Centroid (e) Cityblock / Centroid (f) Minkowski / Centroid

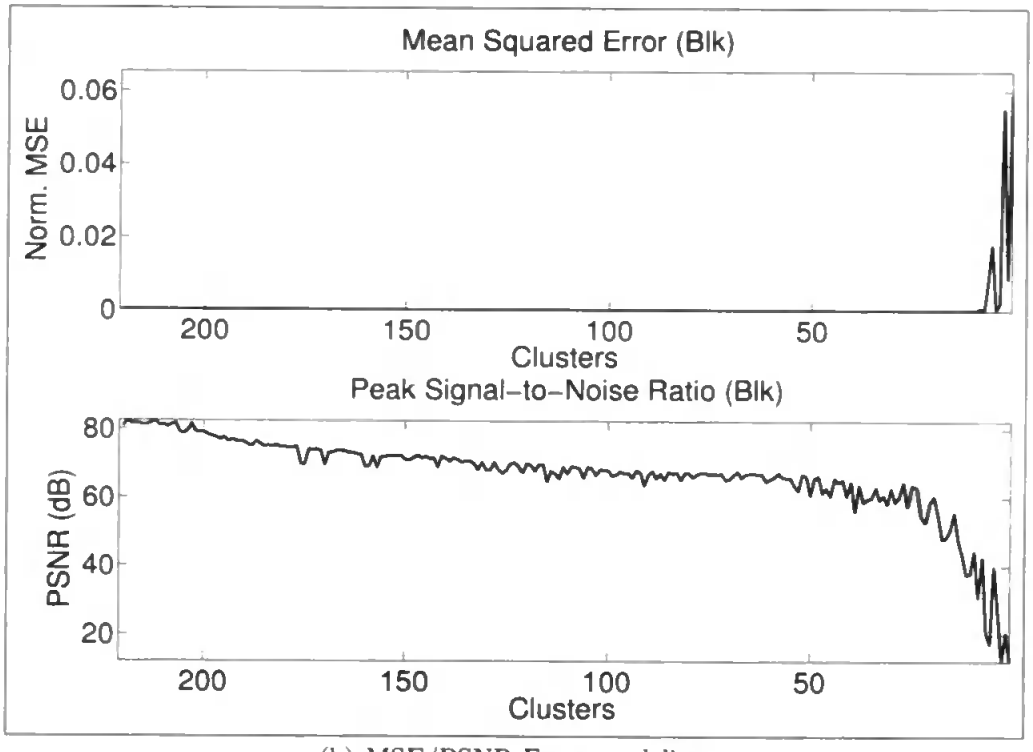
(g) Cosine / Centroid (h) Correlation / Centroid (i) Cityblock / Ward

(j) Minkowski / Ward (k) Cosine / Ward (l) Correlation / Ward

Figure 6.4: Basson - Dendrograms results (*continued*)

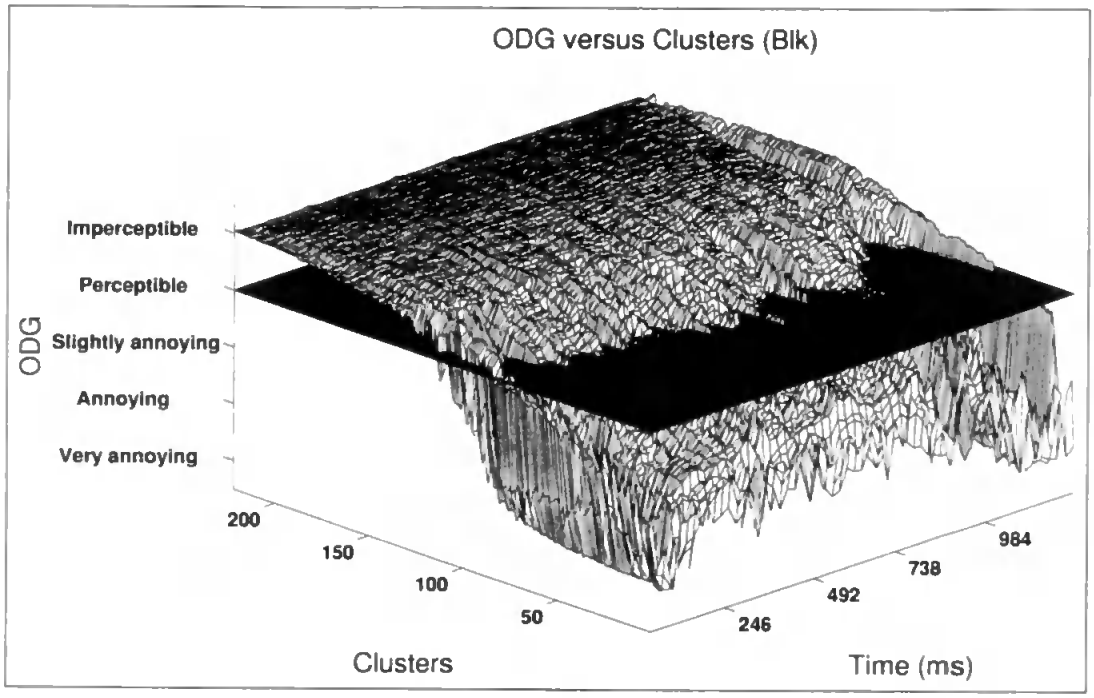


(a) Waveform modeling difference

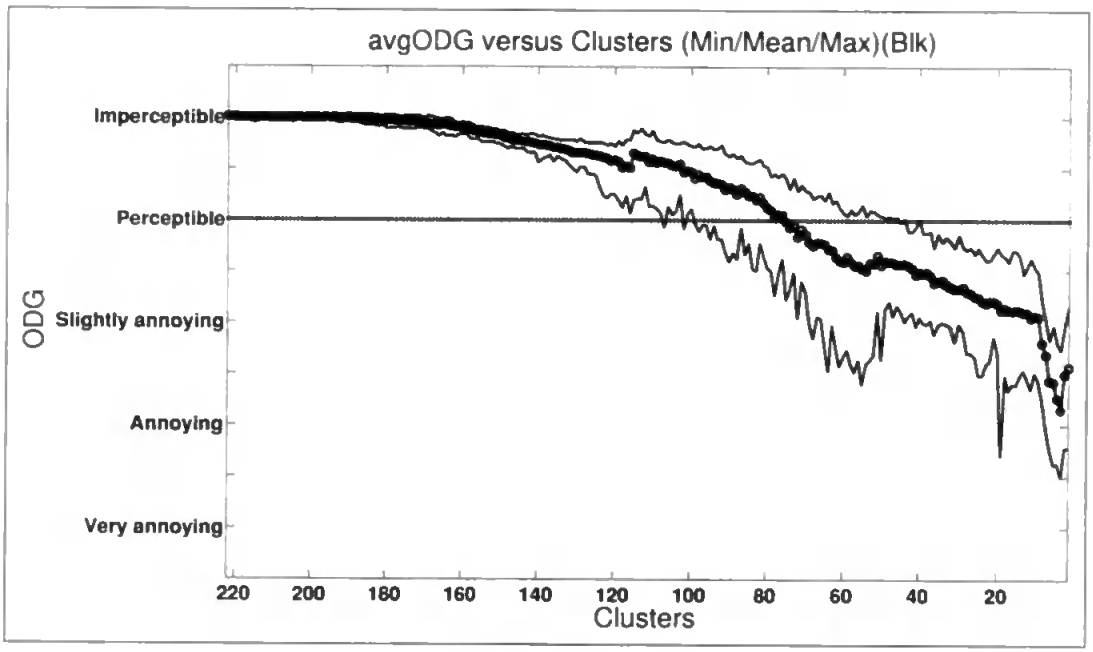


(b) MSE/PSNR Error modeling

Figure 6.5: Basson - Clustering results (Blk)

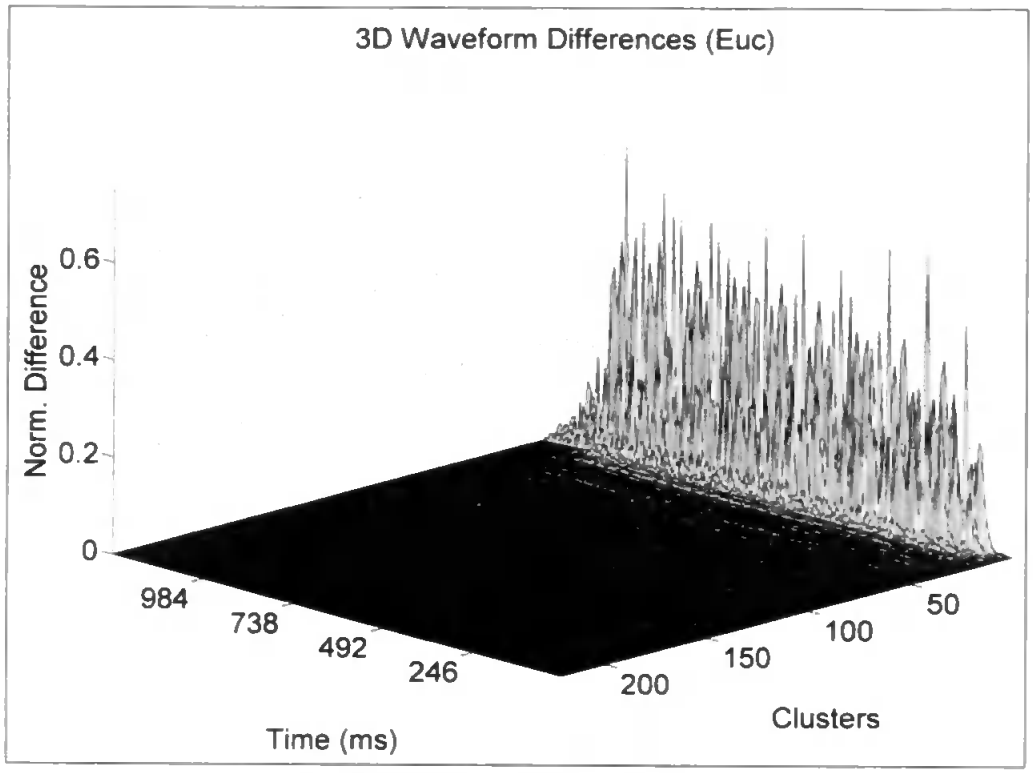


(a) ODGvsCL

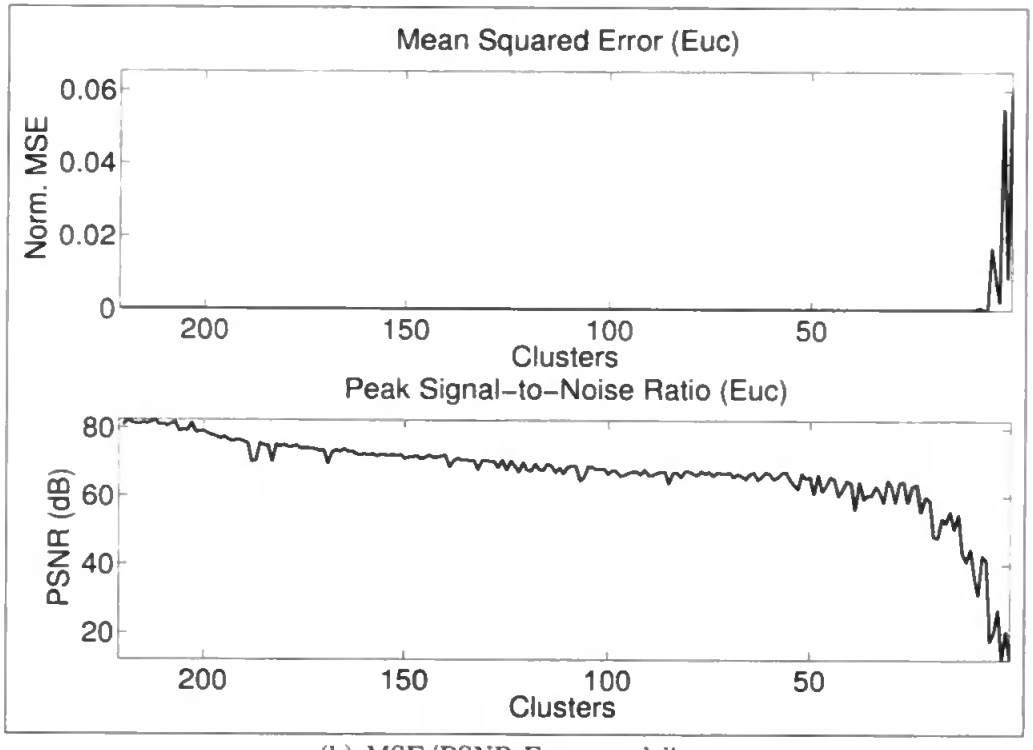


(b) avgODGvsCL

Figure 6.6: Basson - Clustering results ODG (Blk)

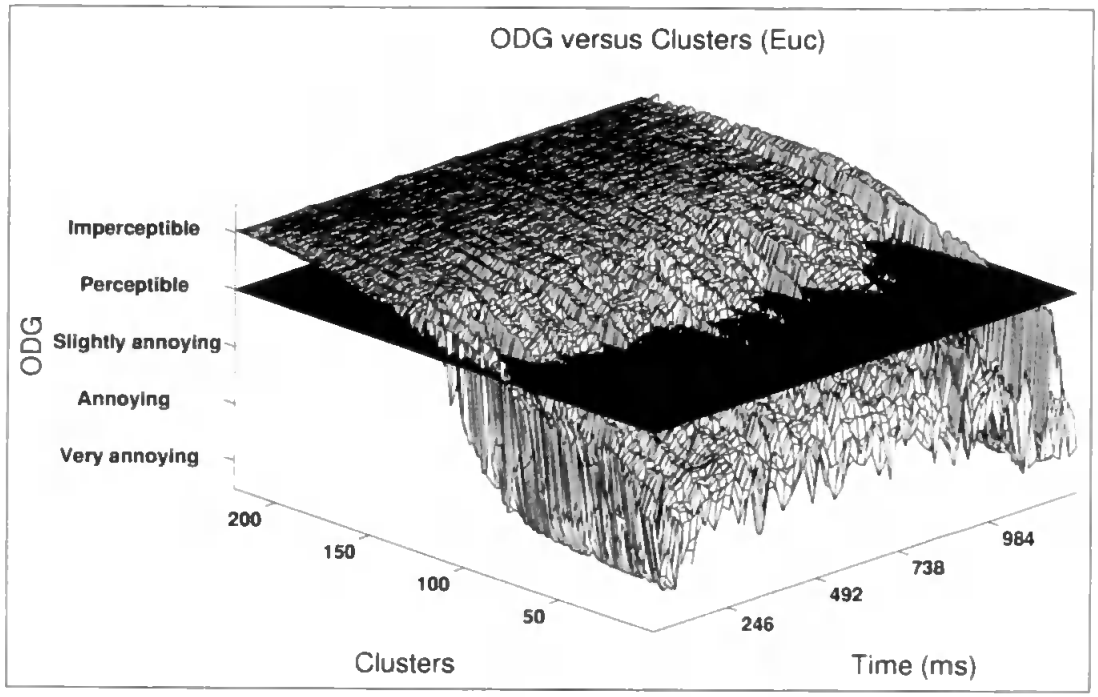


(a) Waveform modeling difference

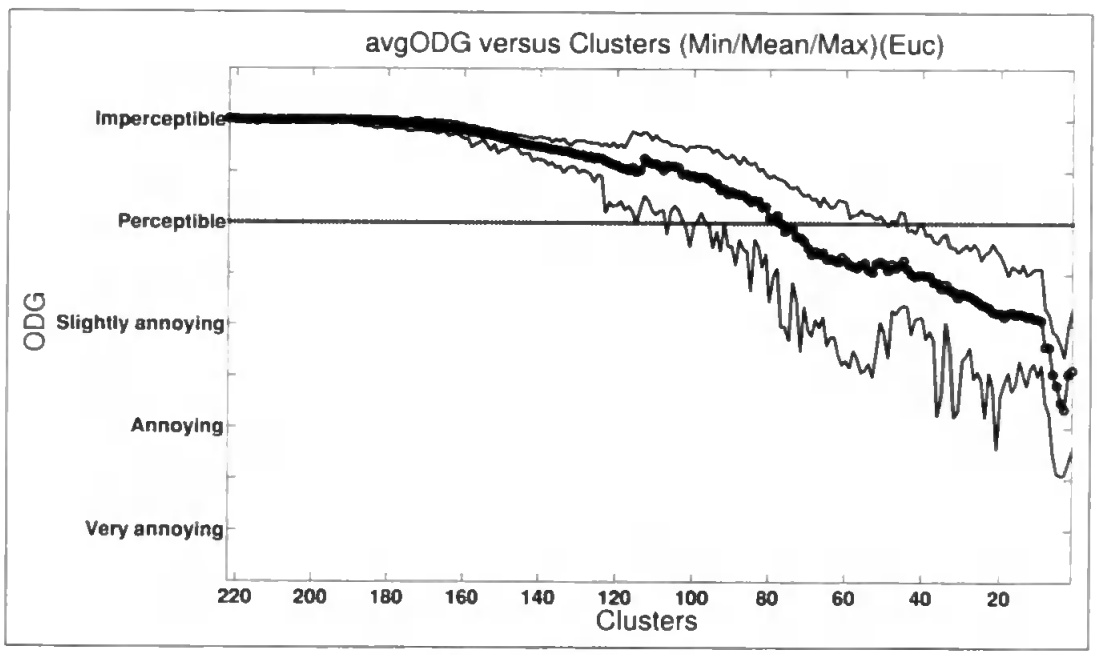


(b) MSE/PSNR Error modeling

Figure 6.7: Basson - Clustering results (Euc)

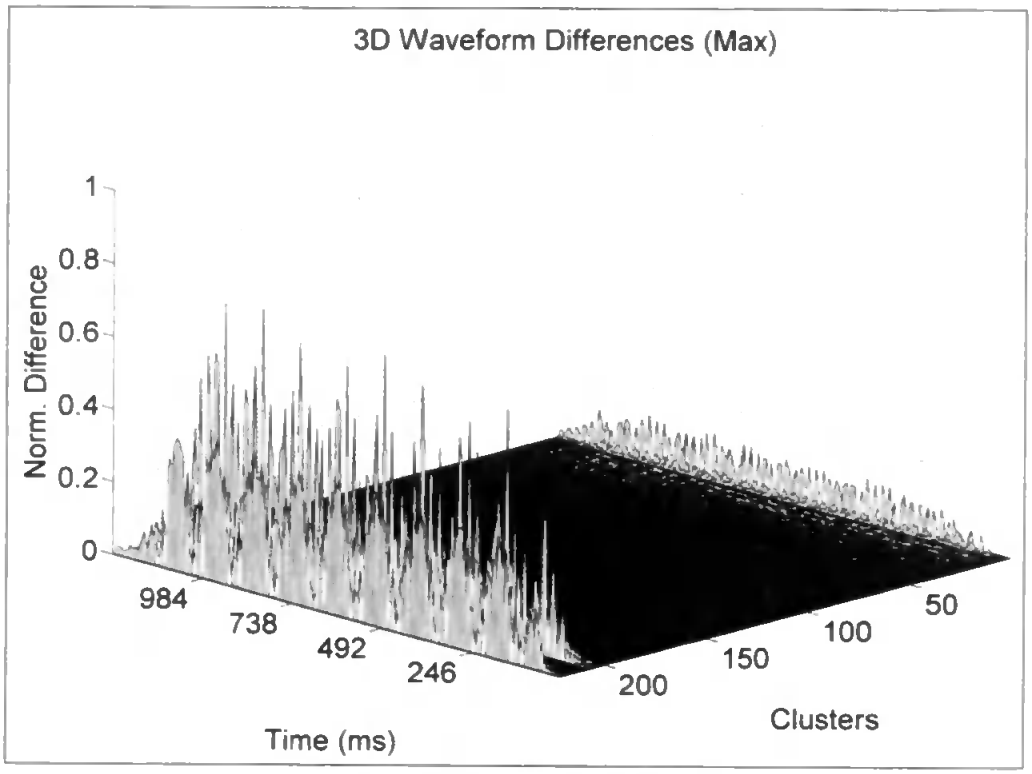


(a) ODGvsCL

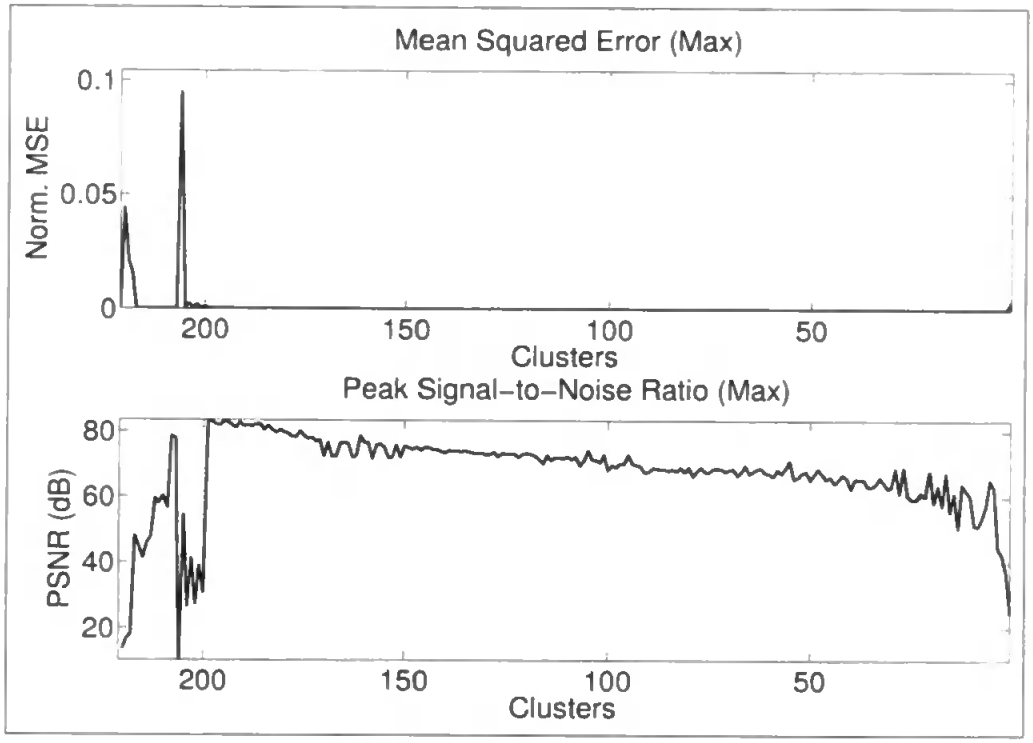


(b) avgODGvsCL

Figure 6.8: Basson - Clustering results ODG (Euc)

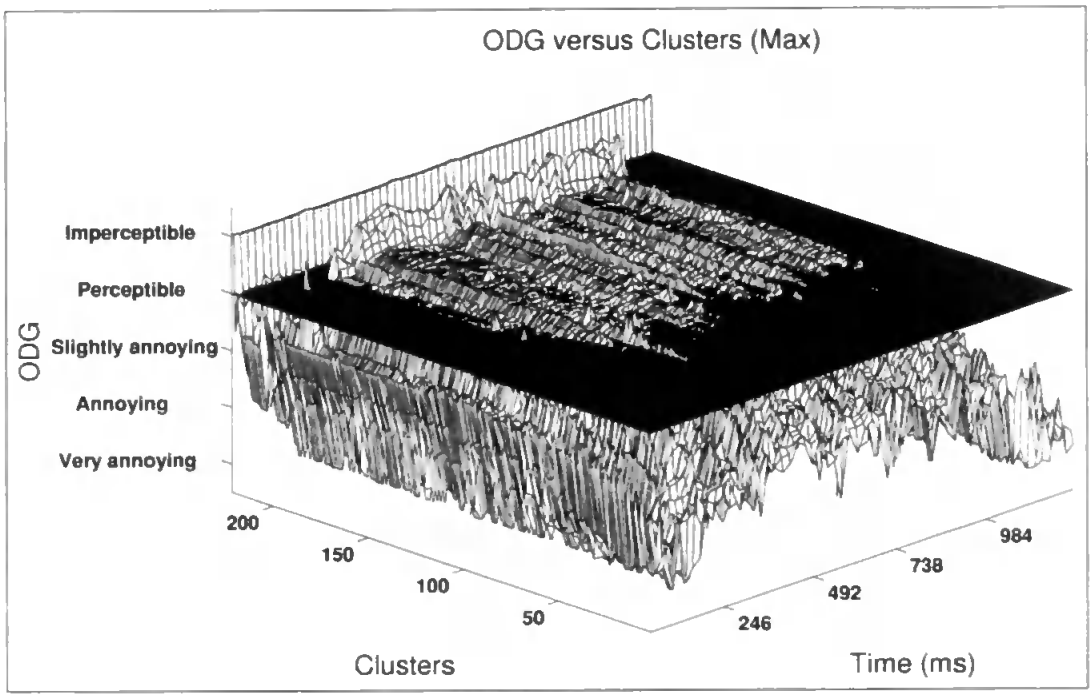


(a) Waveform modeling difference

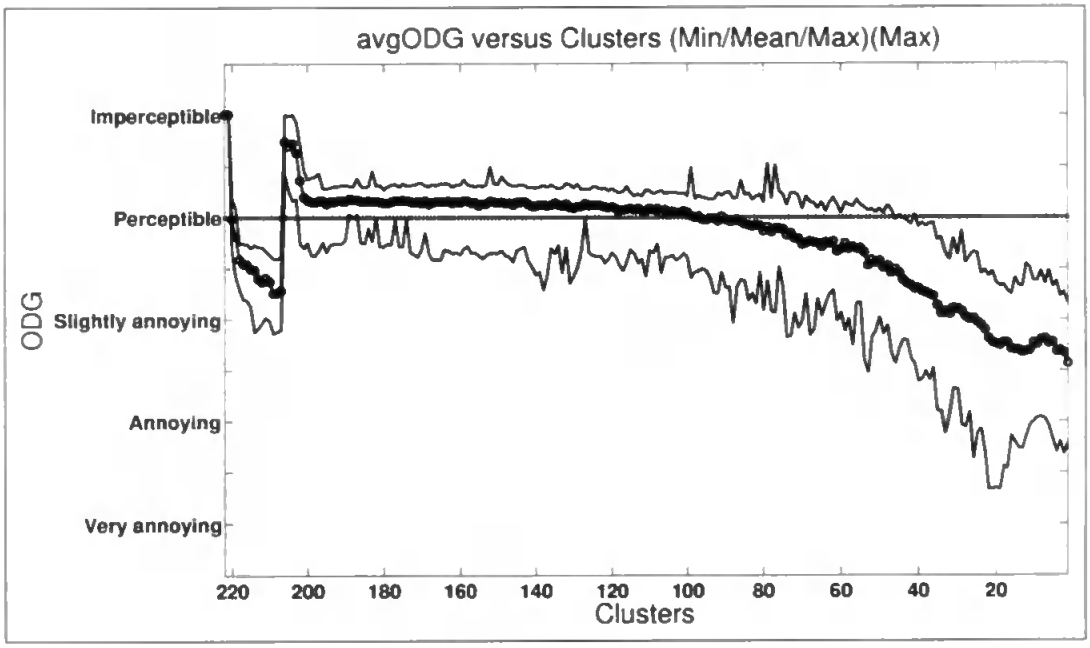


(b) MSE/PSNR Error modeling

Figure 6.9: Basson - Clustering results (Max)

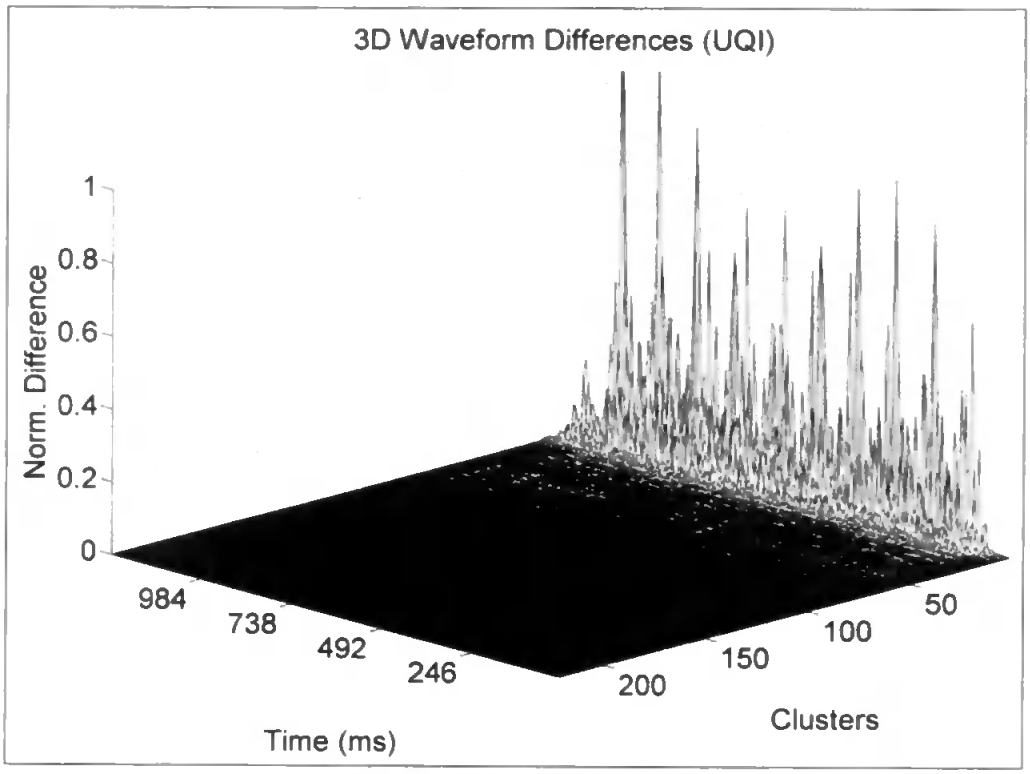


(a) ODGvsCL

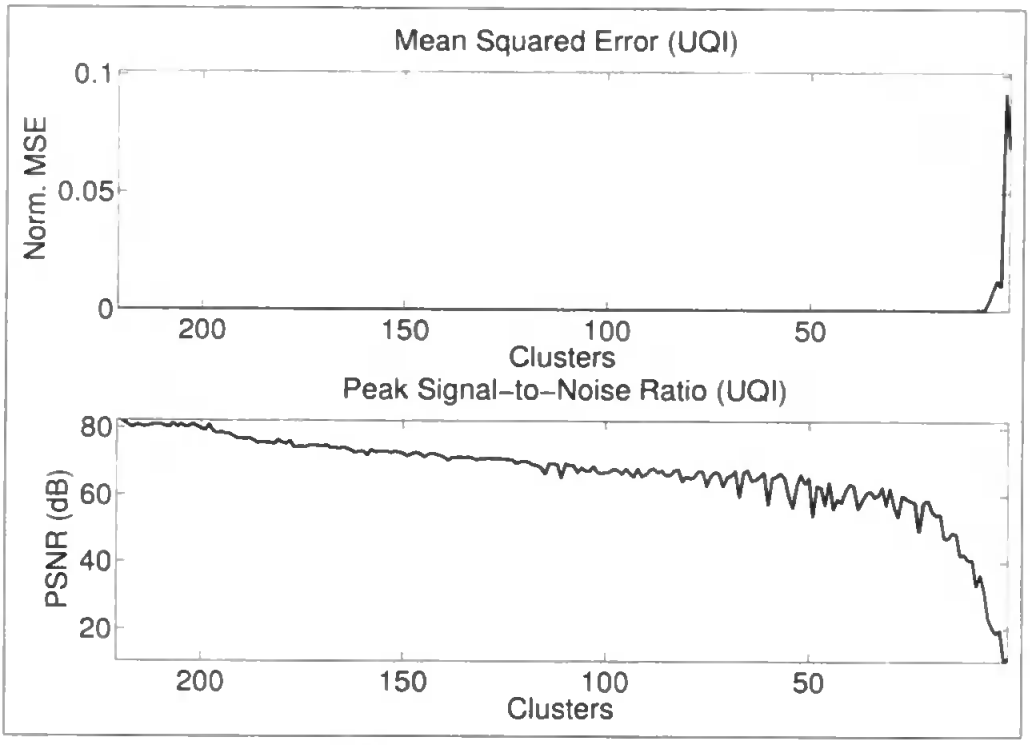


(b) avgODGvsCL

Figure 6.10: Basson - Clustering results ODG (Max)

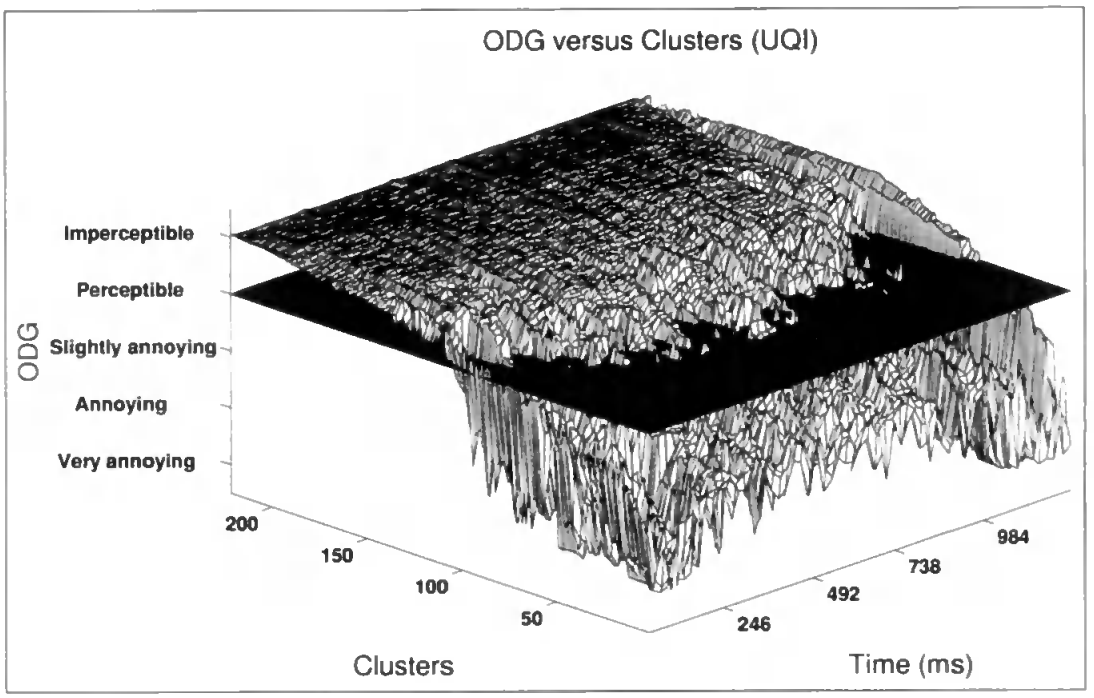


(a) Waveform modeling difference

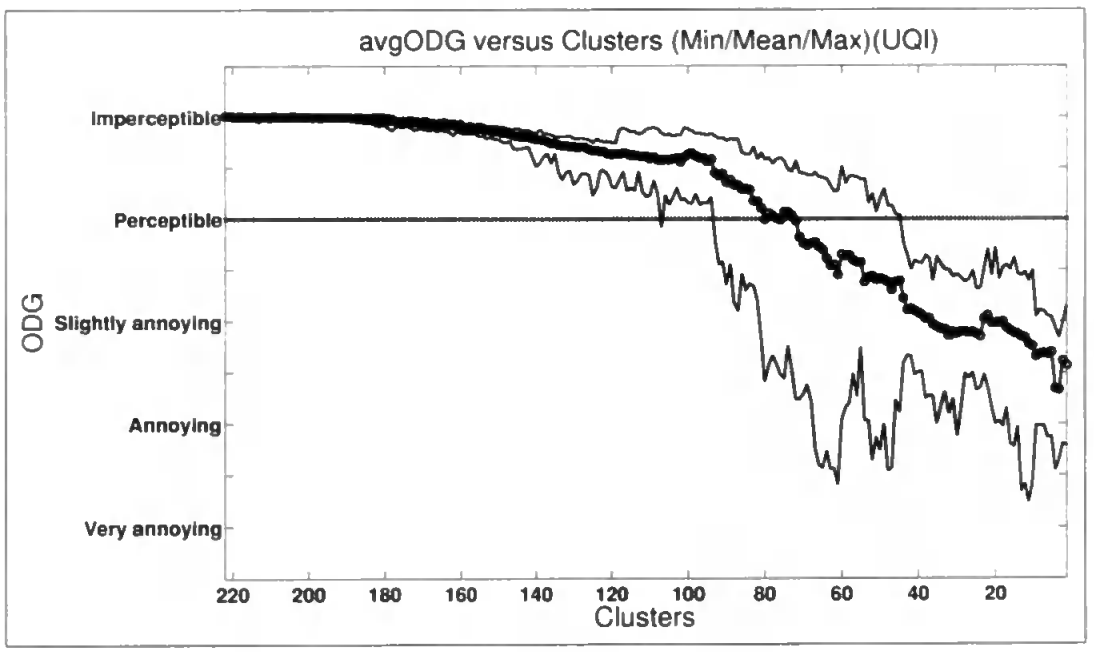


(b) MSE/PSNR Error modeling

Figure 6.11: Basson - Clustering results (UQI)



(a) ODGvsCL



(b) avgODGvsCL

Figure 6.12: Basson - Clustering results ODG (UQI)

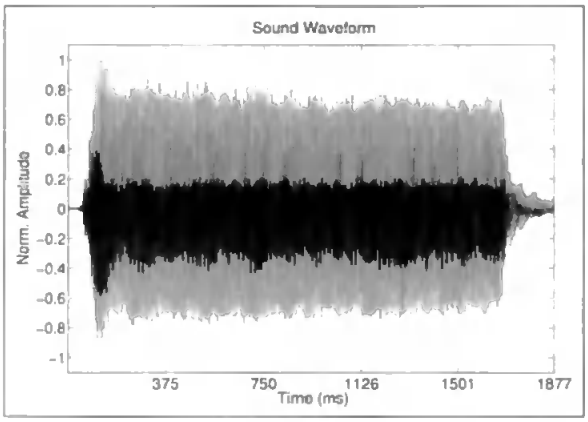
6.4.3 Sound - Dulciana 16'

Sound Analysis - The fundamental is about 131 Hz which results in a large number of bands for the Phase Vocoder analysis. Figure 6.13(a) shows the waveform of the sound, Figure 6.13(c) its spectrum as lines with Figure 6.13(e) showing all the harmonics super-imposed. Figure 6.13(f) shows harmonic distribution with interesting points the highest harmonics is the second and looking a the difference between the sustain level and maximum level it indicate some clear overshoot, and for the case of the 3^{rd} harmonic as well, with both of them presenting a lot of amplitude modulation in the sustain portion.

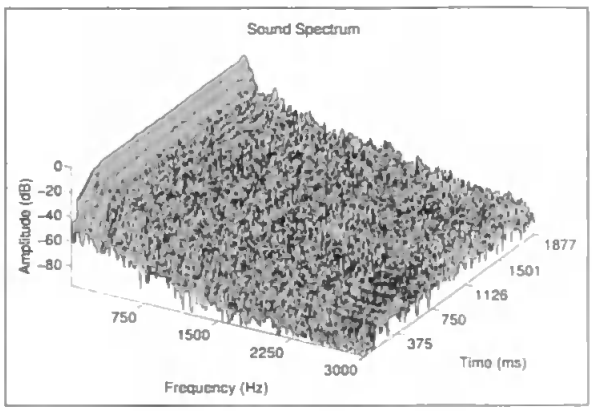
Dendrogram - Figures 6.14(a) to 6.15(l) and Figures 6.15(a) to 6.15(l) show results from the clustering algorithm with the different methods for calculating the metric distance and linkage methods. The Dendrograms have different shapes which are recognisable as *spread* like in Figure 6.3(j), 6.4(c) and 6.4(k), or *rising* like in Figure 6.3(d) and 6.4(g). The most interesting dendrograms seem to be the ones using the correlation as they have a *spread* shape, however one should notice that as the harmonic axis is not ordered it does not correspond to that same shape for all the results. Only a ordered Dendrogram could allow a fair comparison of the dendrograms (See Section 6.6.2).

Modeling Error - The clustering error results, using the Cityblock, Euclidean and UQI distance metric methods are shown in Figure 6.16, Figure 6.18, and Figure 6.20. The modeling error shown in these figures are revealing a more severe in the beginning of the sound portion. This can indicate some potential weakness in the clustering algorithm.

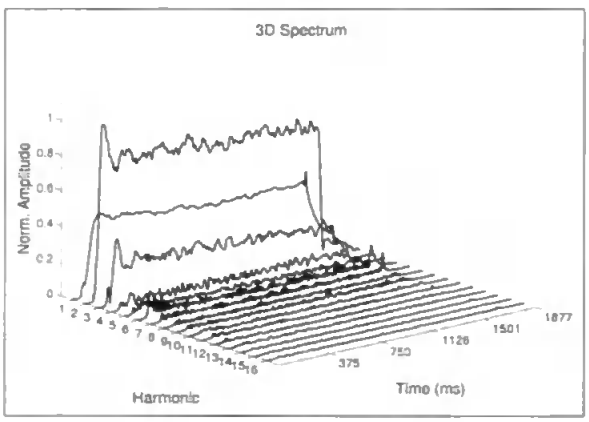
Objective Audio Quality - The objective audio quality results, using the Cityblock, Euclidean and UQI distance metric methods are shown in Figure 6.17, Figure 6.19, and Figure 6.21. The Cityblock and Euclidean are showing similar results with an interesting drop (at clusters number 161) for the case of the Cityblock method. The UQI results present some slightly different shapes.



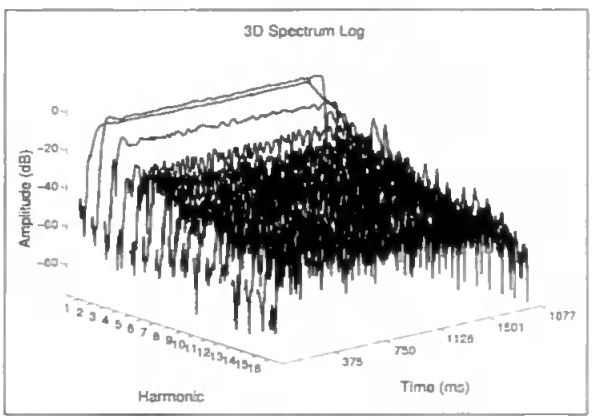
(a) Waveform



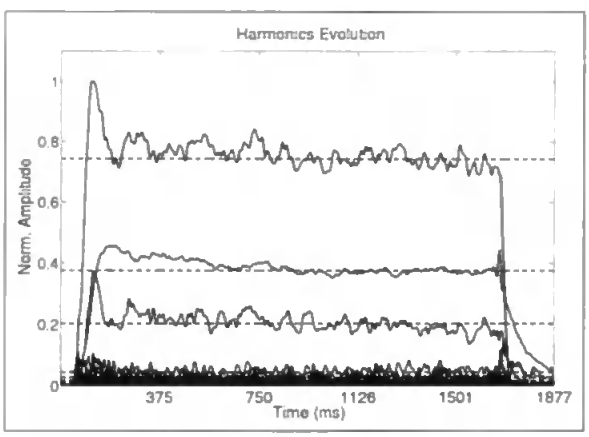
(b) Spectrum



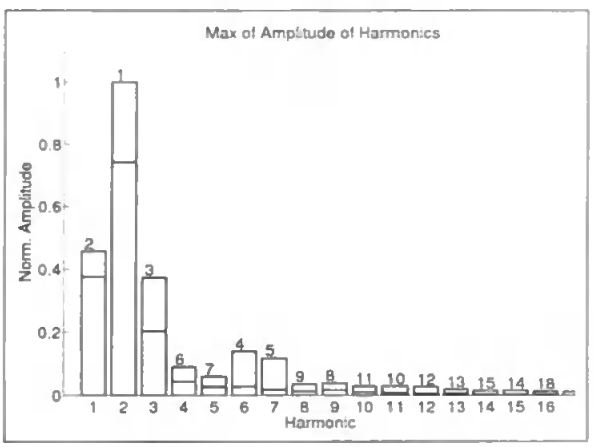
(c) Spectrum lines (Lin)



(d) Spectrum lines (Log)

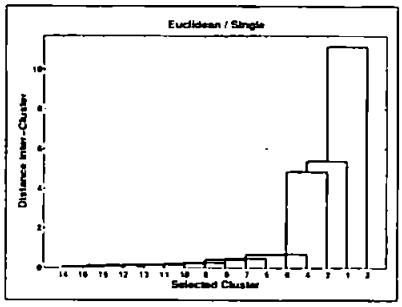


(e) Harmonics envelopes

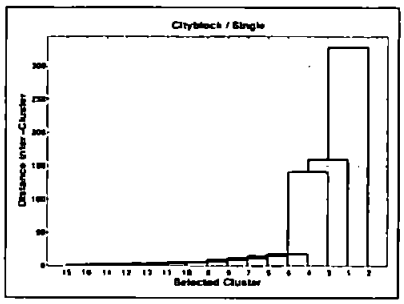


(f) Harmonics distribution

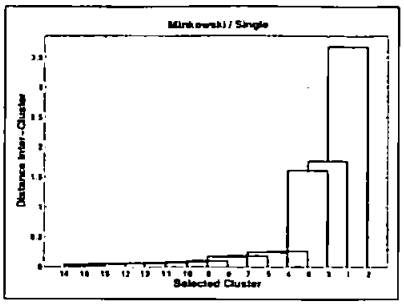
Figure 6.13: Dulciana - Sound analysis



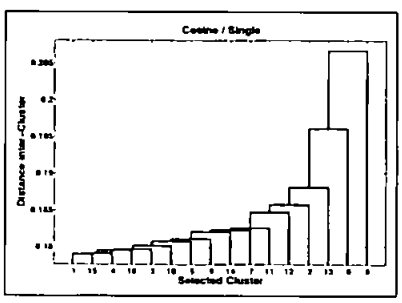
(a) Euclidean / Single



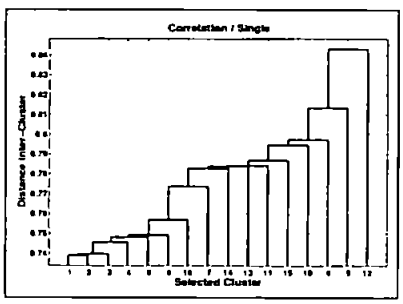
(b) Cityblock / Single



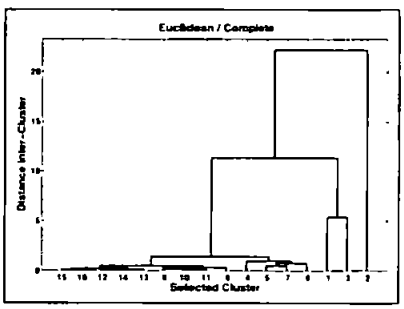
(c) Minkowski / Single



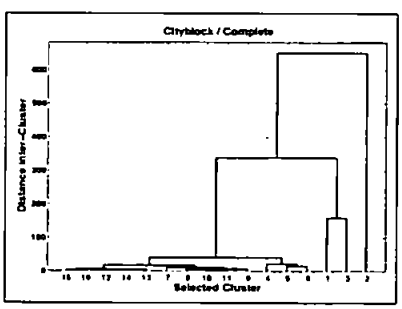
(d) Cosine / Single



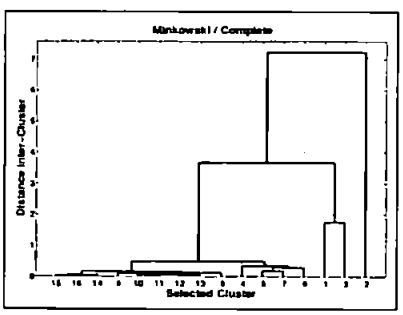
(e) Correlation / Single



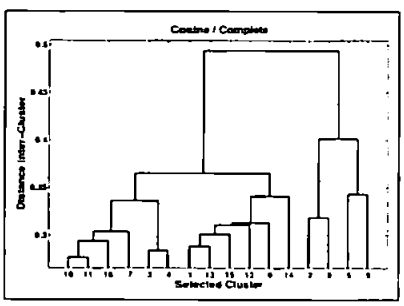
(f) Euclidean / Complete



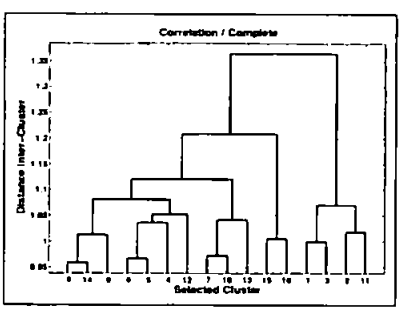
(g) Cityblock / Complete



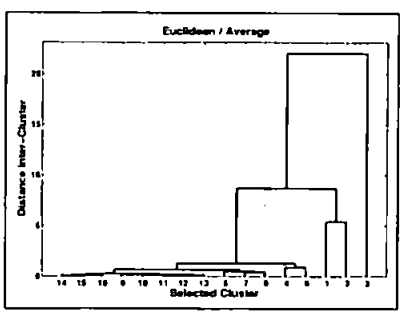
(h) Minkowski / Complete



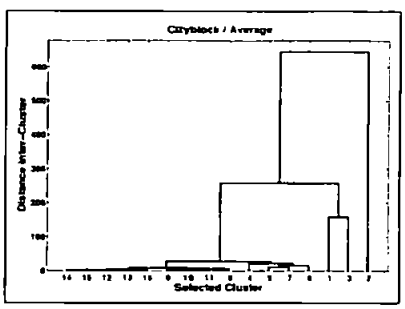
(i) Cosine / Complete



(j) Correlation / Complete

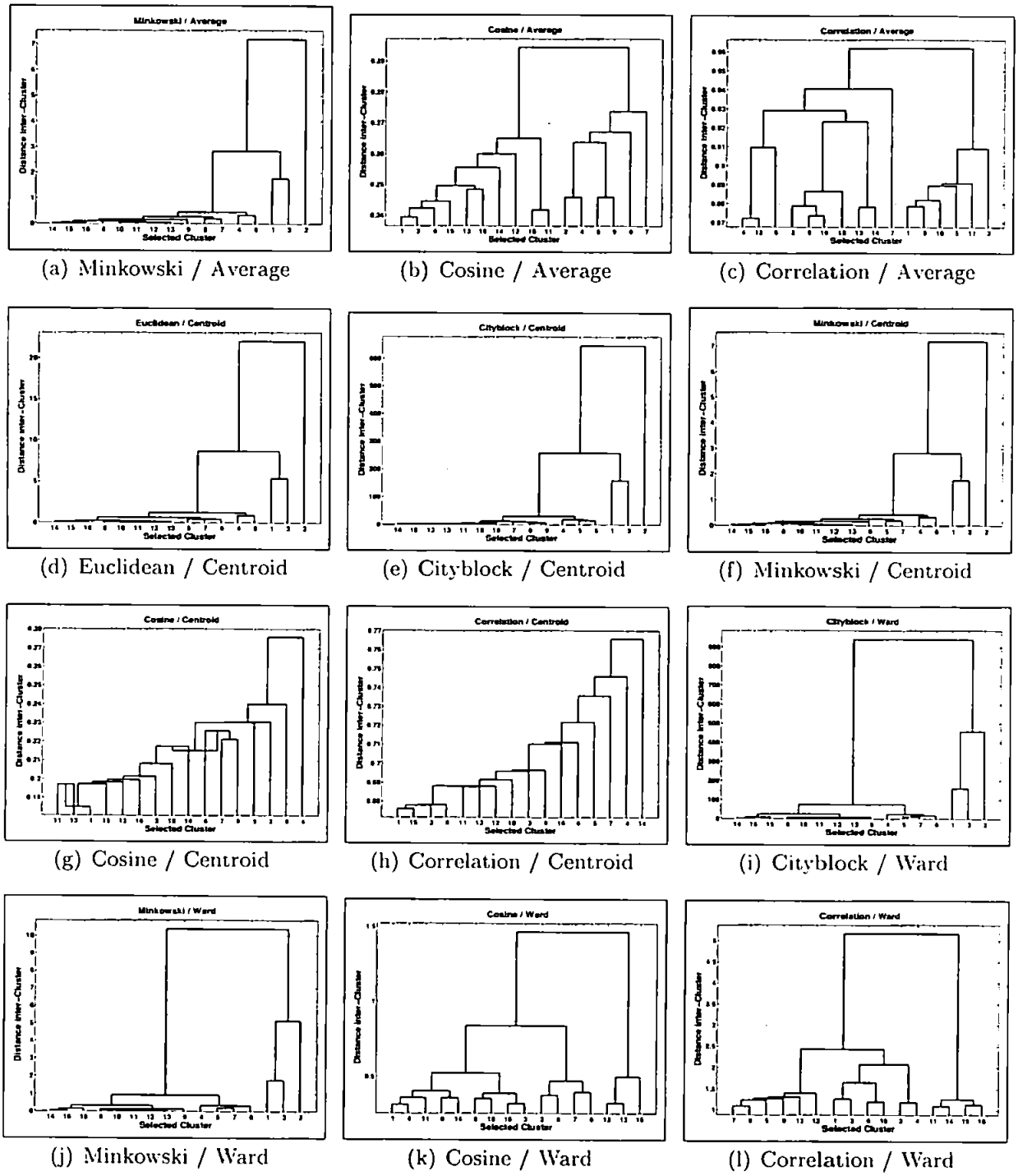


(k) Euclidean / Average



(l) Cityblock / Average

Figure 6.14: Dulciana - Dendrograms results



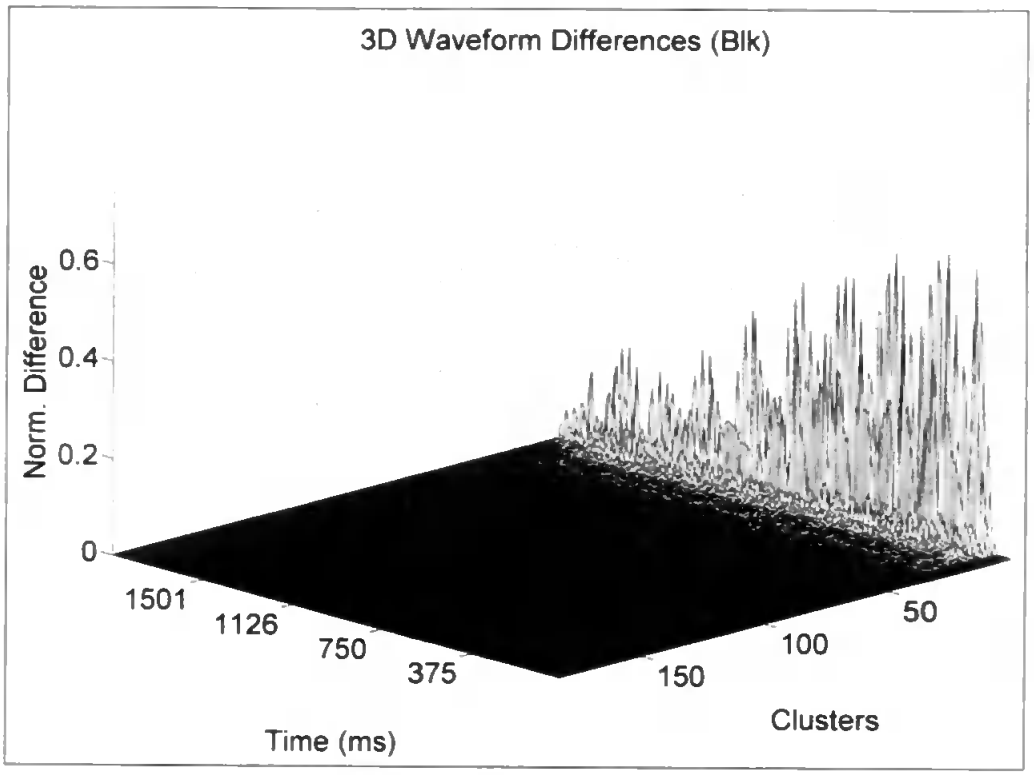
(a) Minkowski / Average (b) Cosine / Average (c) Correlation / Average

(d) Euclidean / Centroid (e) Cityblock / Centroid (f) Minkowski / Centroid

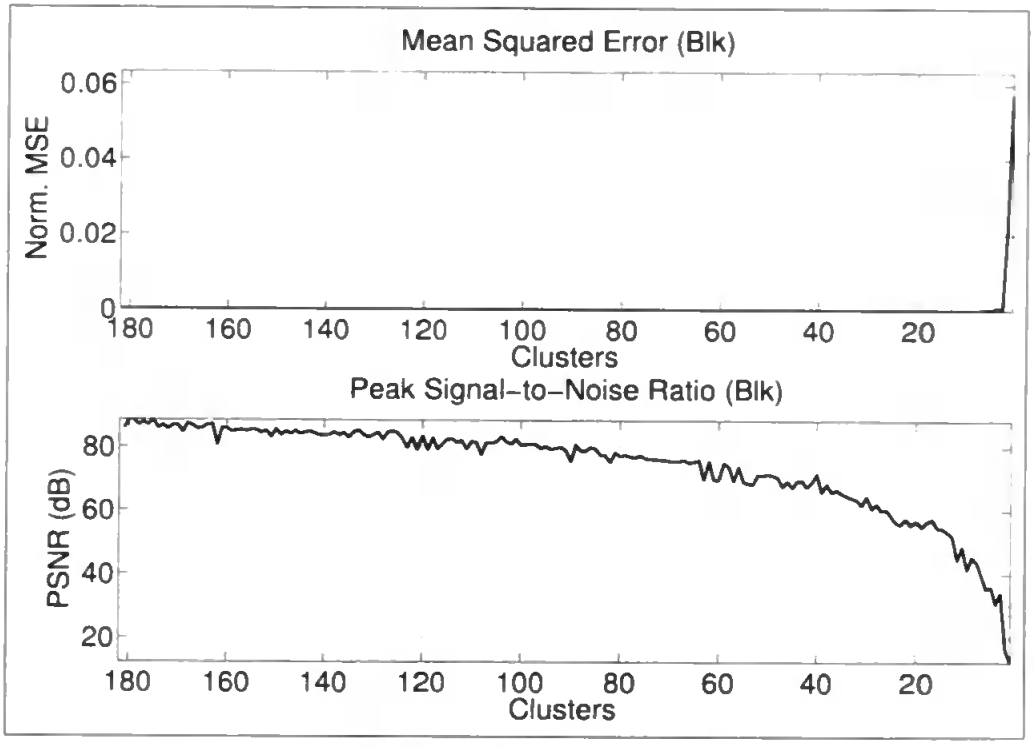
(g) Cosine / Centroid (h) Correlation / Centroid (i) Cityblock / Ward

(j) Minkowski / Ward (k) Cosine / Ward (l) Correlation / Ward

Figure 6.15: Dulciana - Dendrograms results (*continued*)

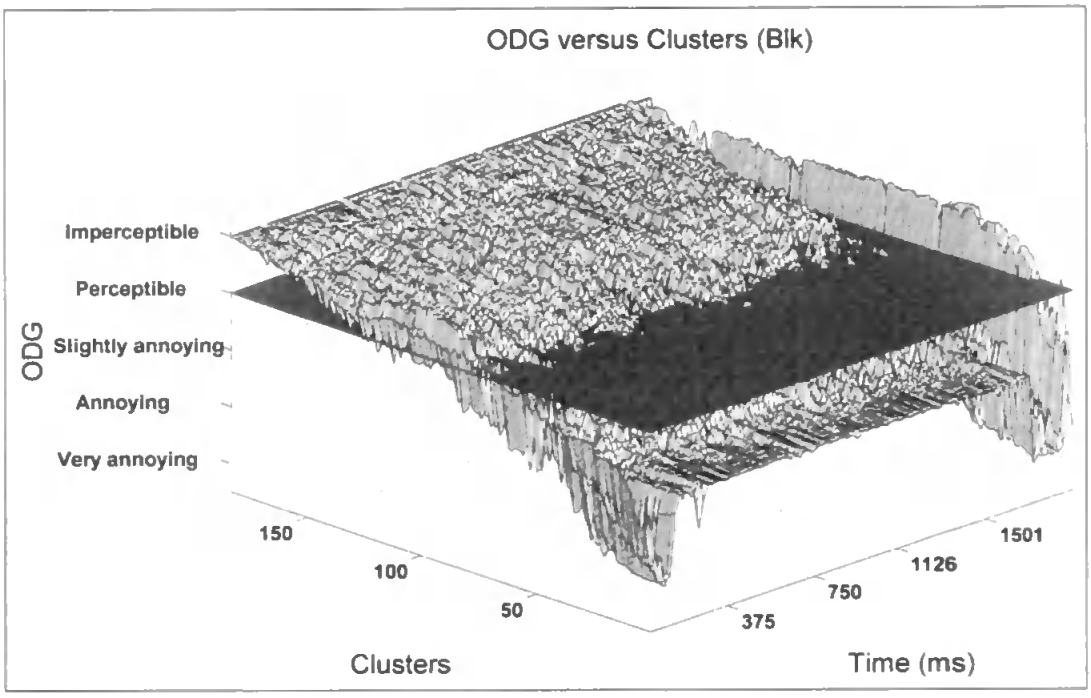


(a) Waveform modeling difference

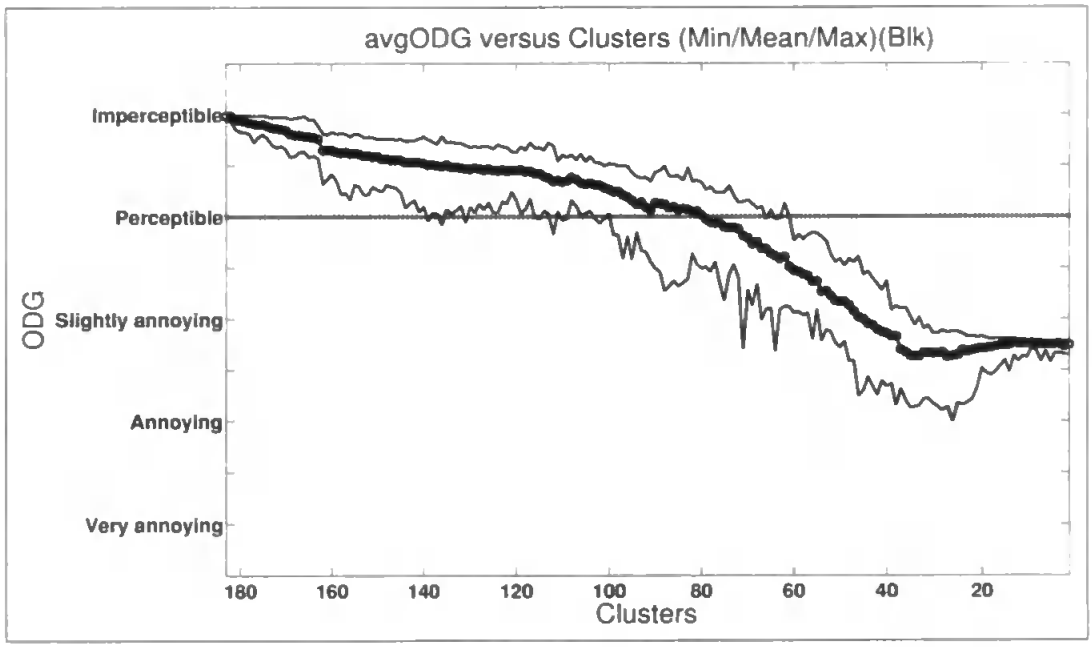


(b) MSE/PSNR Error modeling

Figure 6.16: Dulciana - Clustering results (Blk)

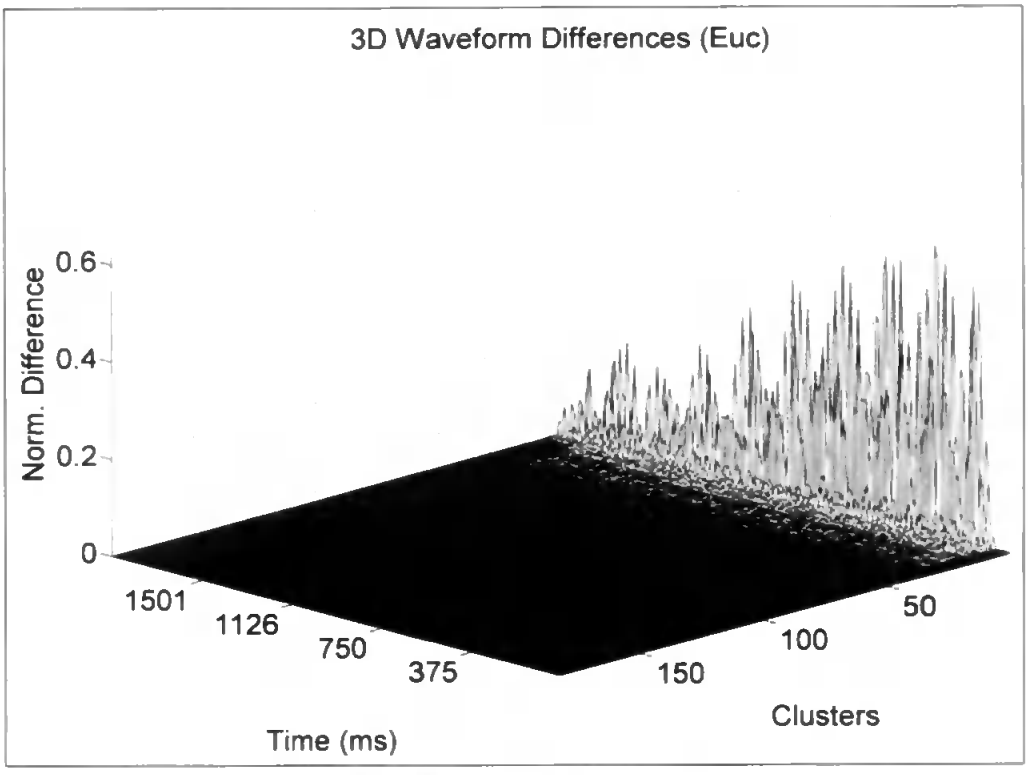


(a) ODGvsCL

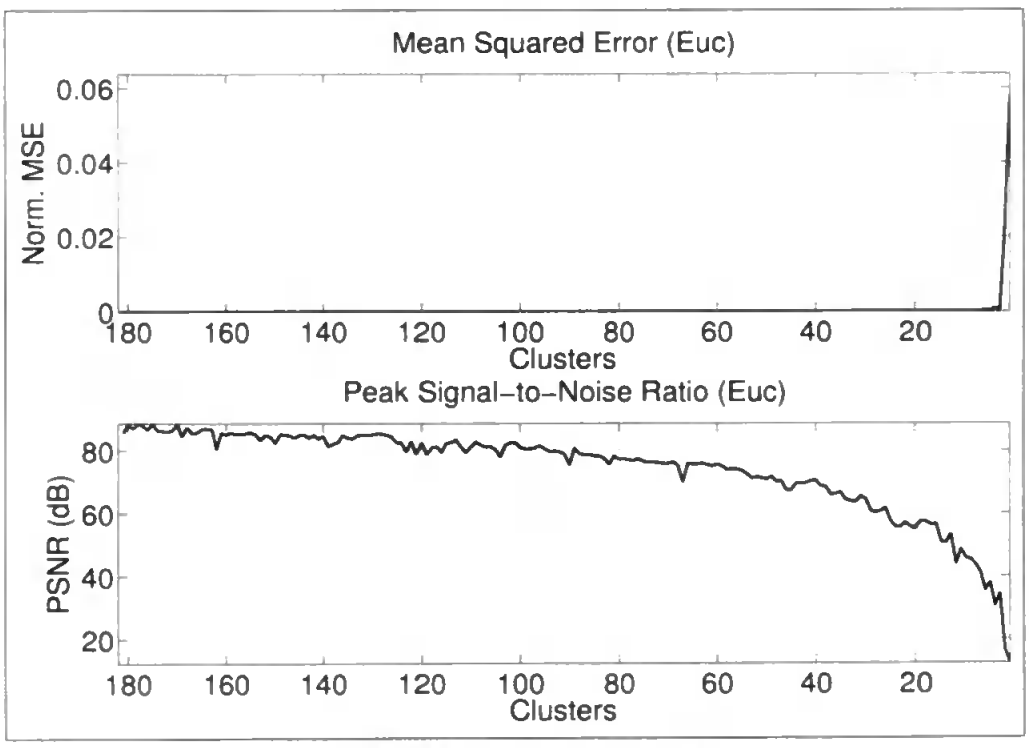


(b) avgODGvsCL

Figure 6.17: Dulciana - Clustering results ODG (Blk)

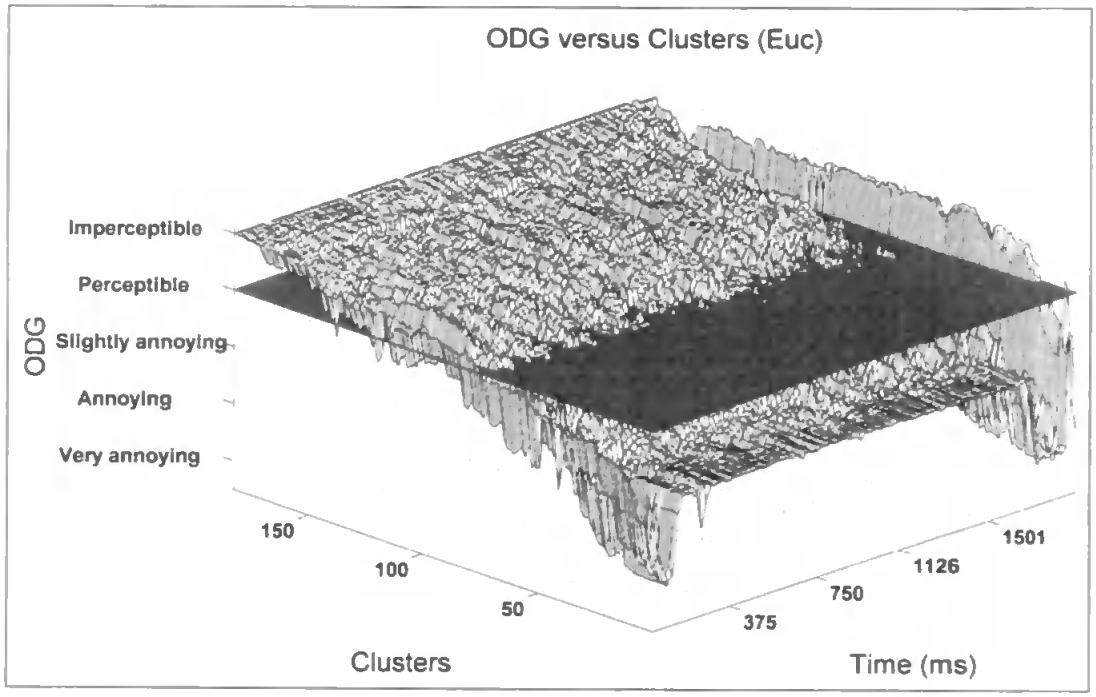


(a) Waveform modeling difference

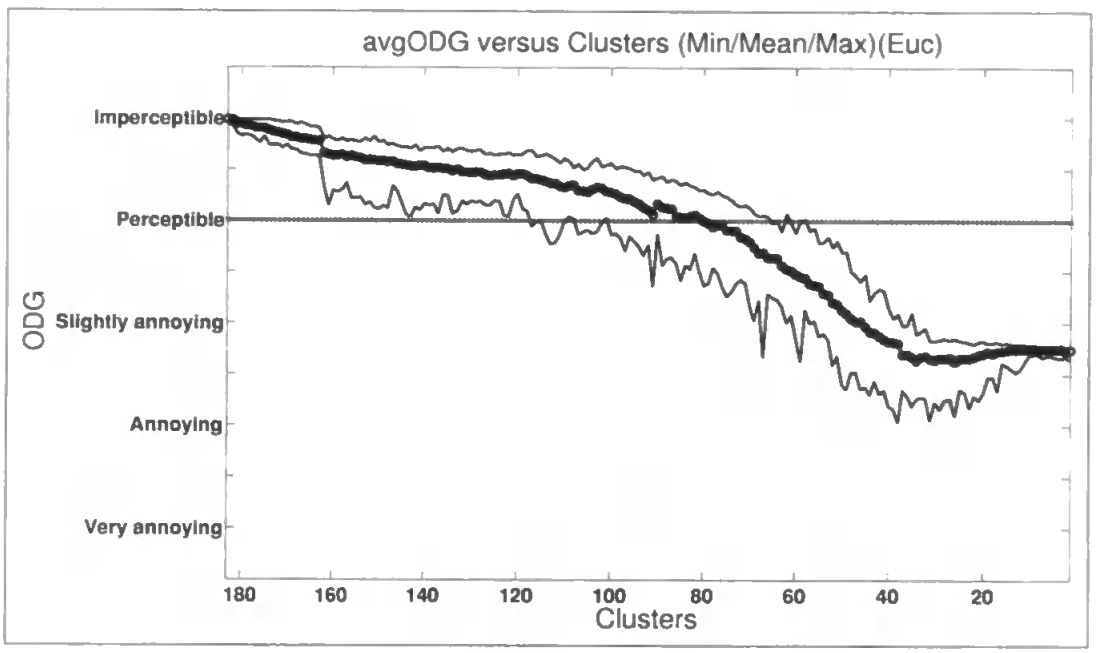


(b) MSE/PSNR Error modeling

Figure 6.18: Dulciana - Clustering results (Euc)

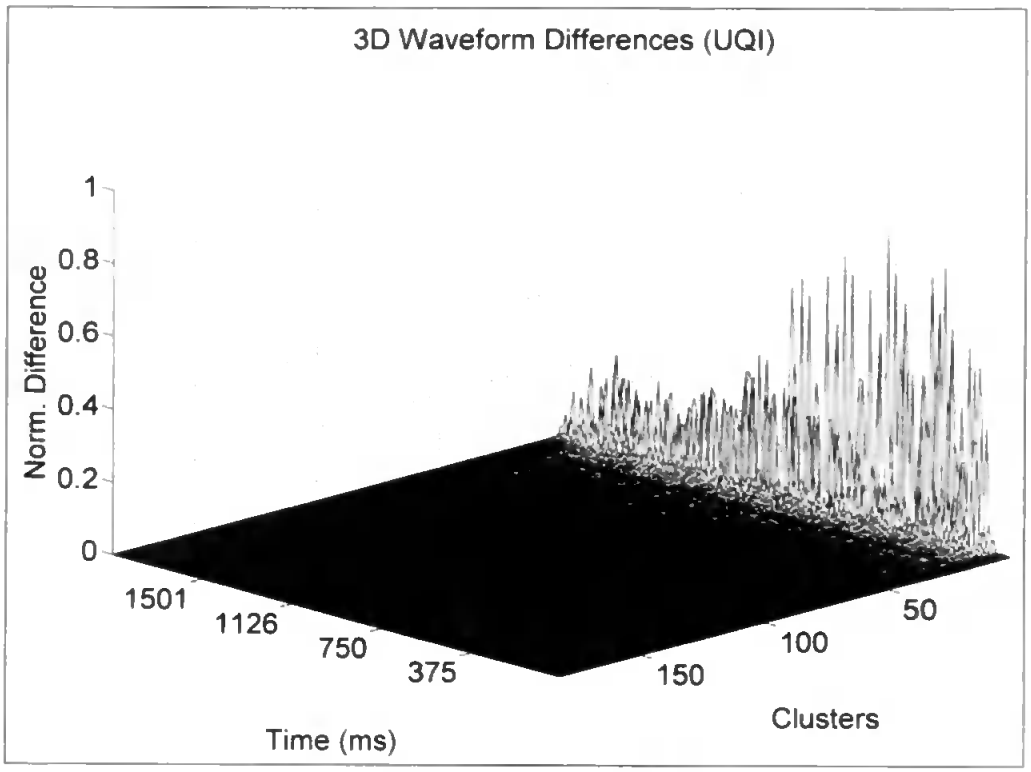


(a) ODGvsCL

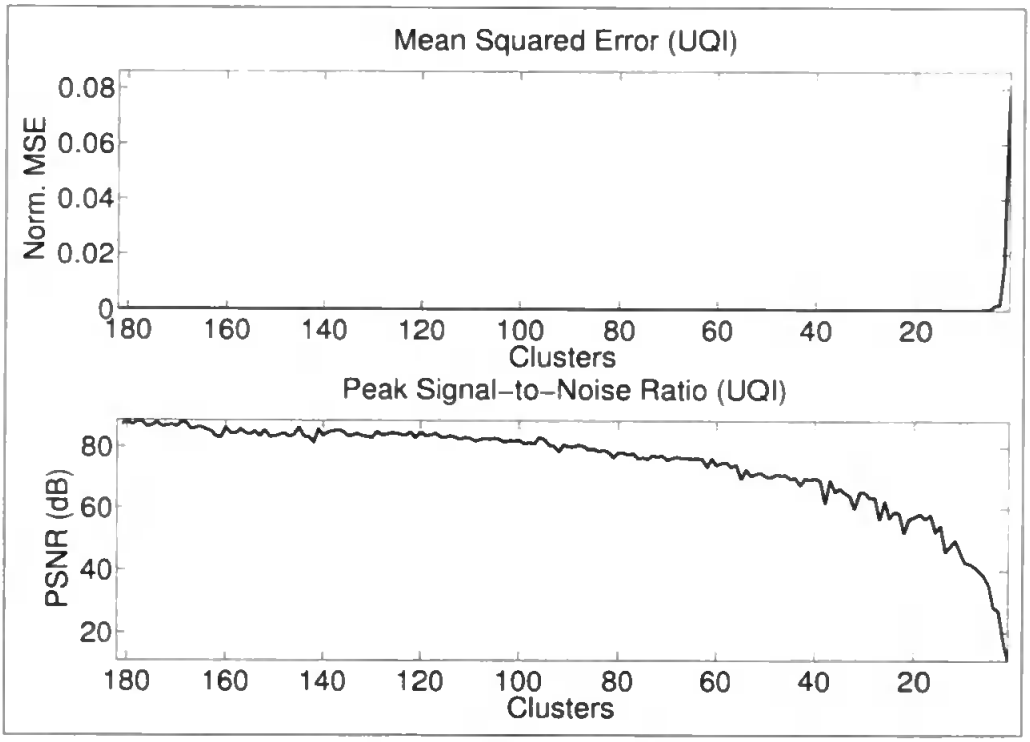


(b) avgODGvsCL

Figure 6.19: Dulciana - Clustering results ODG (Euc)

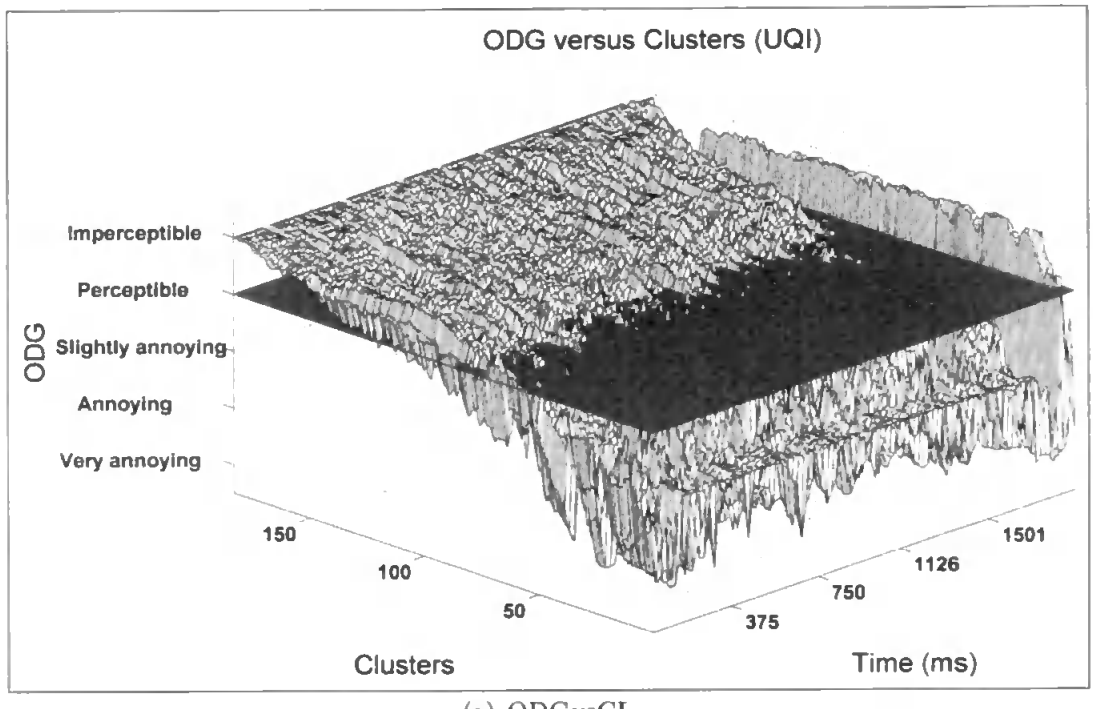


(a) Waveform modeling difference

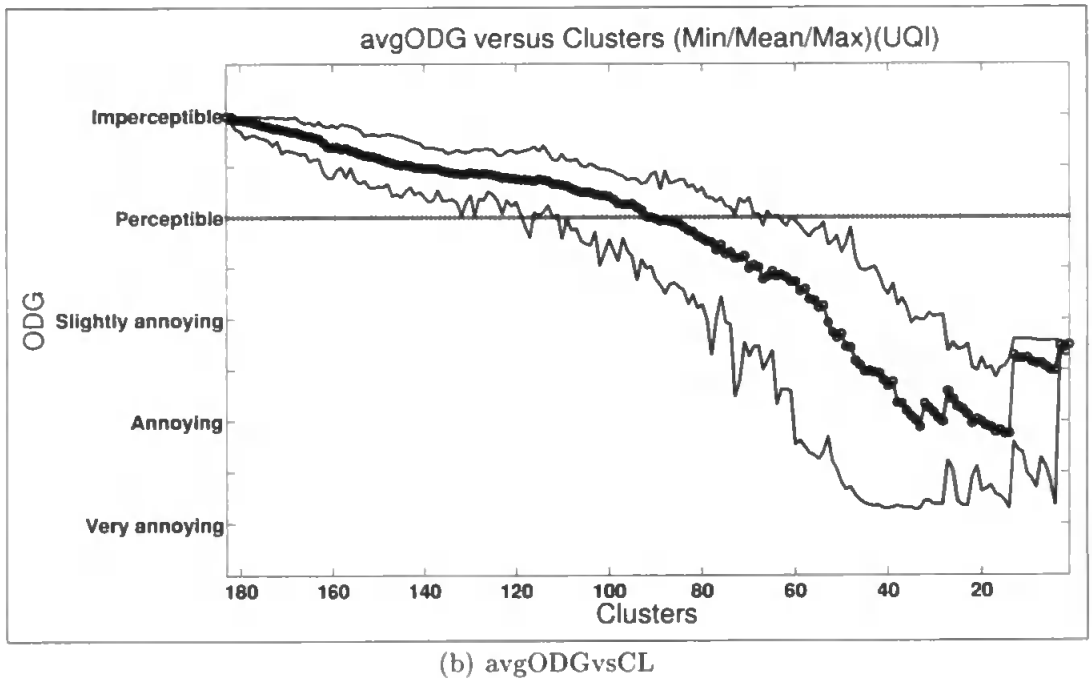


(b) MSE/PSNR Error modeling

Figure 6.20: Dulciana - Clustering results (UQI)



(a) ODGvsCL



(b) avgODGvsCL

Figure 6.21: Dulciana - Clustering results ODG (UQI)

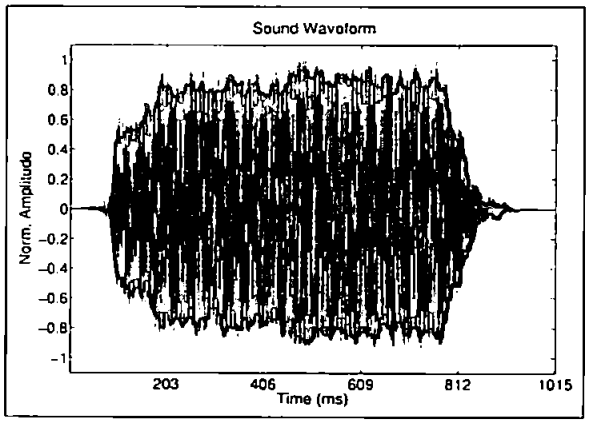
6.4.4 Sound - Flute

Sound Analysis - The waveform of the Flute sound is shown in Figure 6.22(a). Figure 6.22(b) shows the spectrum, while Figure 6.22(c) and Figure 6.22(d) show that same but with spectrum lines which can help to have a clearer view of the evolution of the harmonic. Figure 6.22(d) shows the noise of the harmonics from the 3^{rd} upwards, as this sound is very breathy. Figure 6.22(e) is showing harmonic and their distribution in in Figure 6.22(f). Flute sounds usually have one predominant harmonic which make their tonal characteristic very easily recognisable. The modeling of Flute sounds is the easiest case of pipe organ sounds.

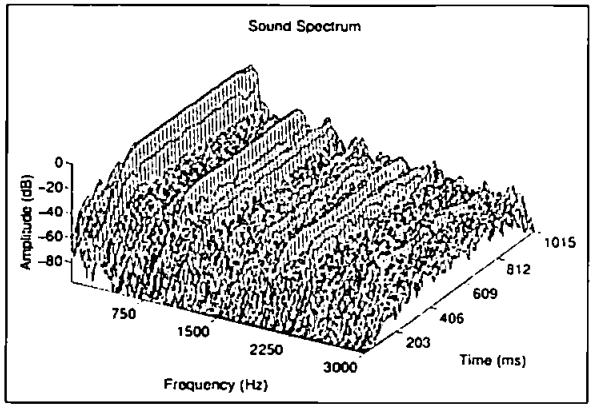
Dendrogram - Figures 6.23(a) to 6.23(l) and Figures 6.24(a) to 6.24(l) show results from the clustering algorithm with the different methods for calculating the metric distance and linkage methods. Figure 6.24(a), 6.24(f) and 6.24(j) shows that Flute sounds have one very dominant harmonic and which one it is looking at the last vertical leg of the dendrograms.

Modeling Error - The clustering error results, using Cityblock, Euclidean, Max and UQI distance metric methods are shown in Figure 6.25, Figure 6.27, Figure 6.29 and Figure 6.31. It is clear that the maximum modeling error for all the Flute modeling cases are much smaller (0.25) than the previous sounds cases (for example 0.7 to 1.0 for Basson), with a concentration in the sustain portion (flat parts at the beginning and end along the time axis)

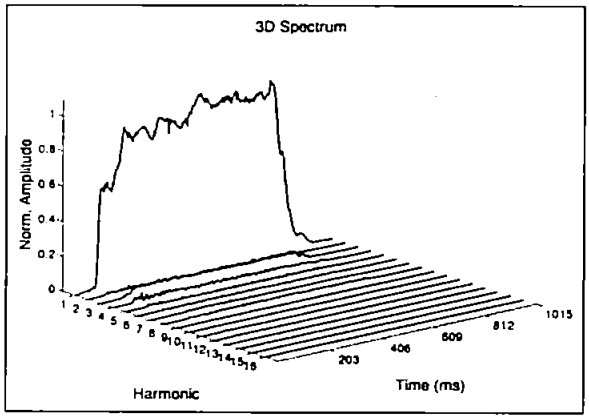
Objective Audio Quality - Figure 6.26, Figure 6.28, Figure 6.30, and Figure 6.32 are showing the objective audio quality results for the flute sound, using the Cityblock, Euclidean, Max and UQI distance metric methods. All results present a very flat beginning part of the objective audio quality surface (and curve) with a *knee* after reducing the harmonic structure by 30 clusters, falling gradually after. The case of UQI provide better results with a mean cutting the perceptible line (See Figure 6.32(b)) at cluster number



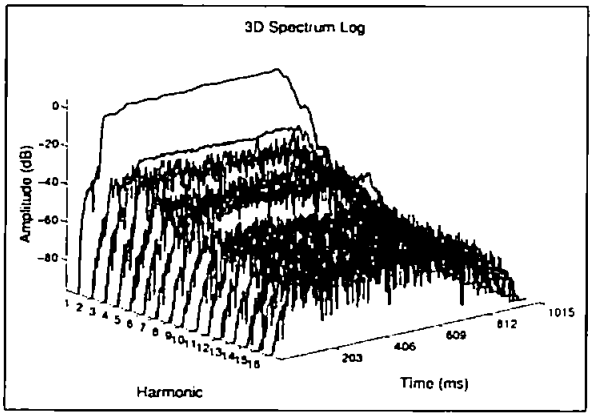
(a) Waveform



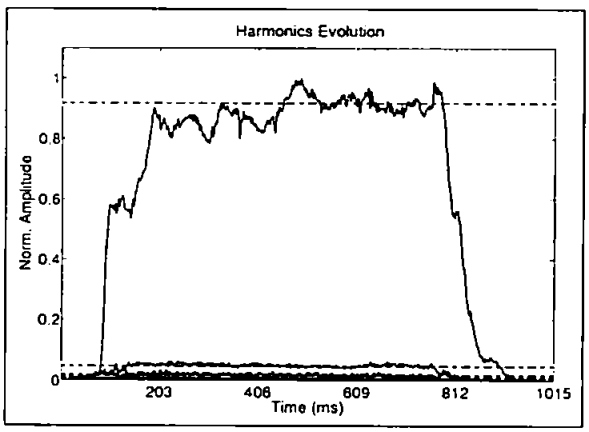
(b) Spectrum



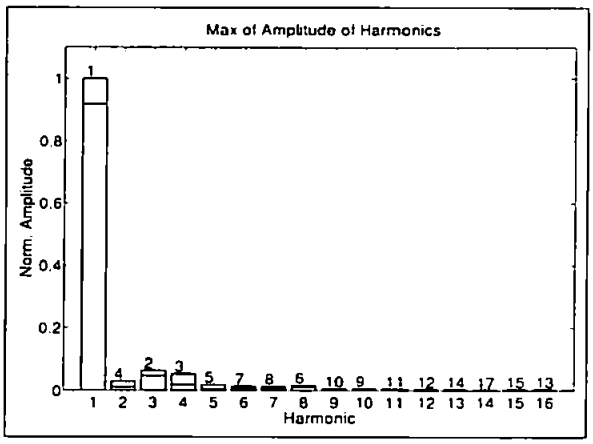
(c) Spectrum lines (Lin)



(d) Spectrum lines (Log)



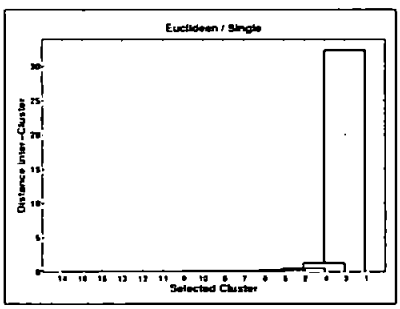
(e) Harmonics envelopes



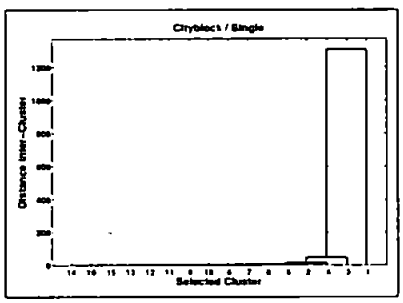
(f) Harmonics distribution

Figure 6.22: Flute - Sound analysis

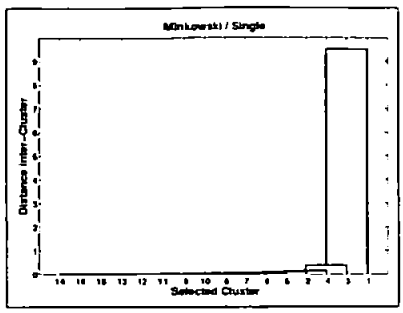
19 instead of 27 for the other cases.



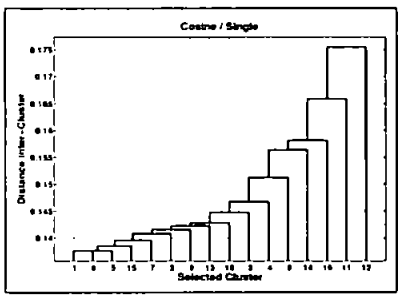
(a) Euclidean / Single



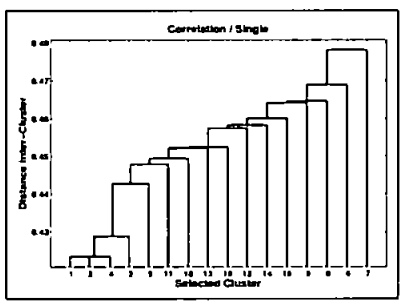
(b) Cityblock / Single



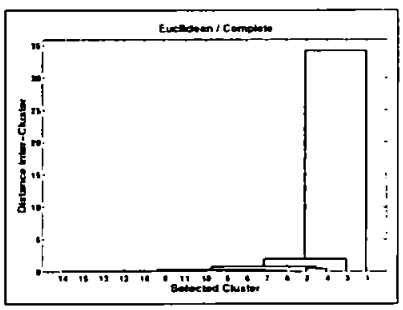
(c) Minkowski / Single



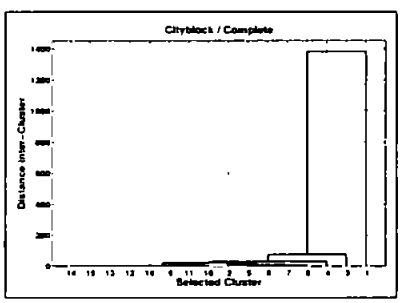
(d) Cosine / Single



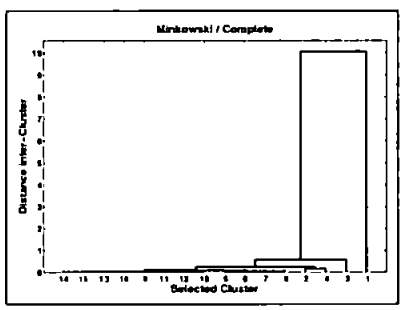
(e) Correlation / Single



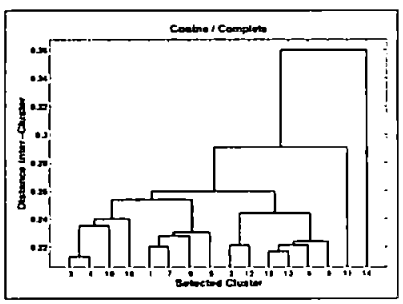
(f) Euclidean / Complete



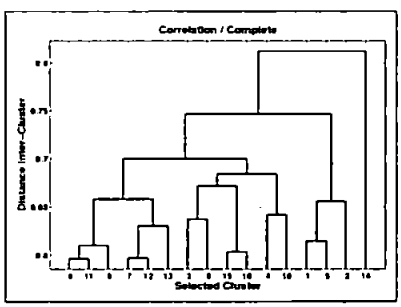
(g) Cityblock / Complete



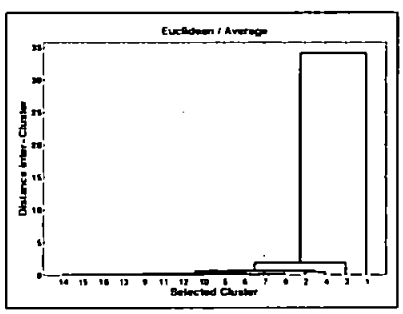
(h) Minkowski / Complete



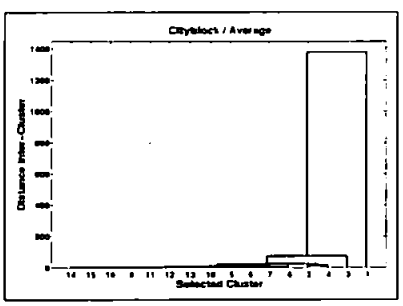
(i) Cosine / Complete



(j) Correlation / Complete

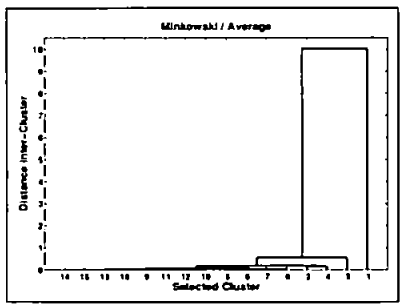


(k) Euclidean / Average

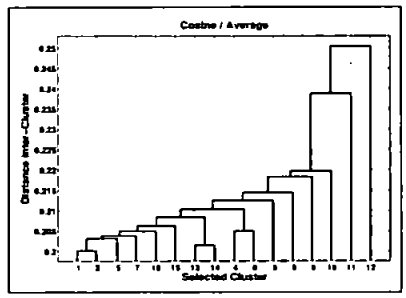


(l) Cityblock / Average

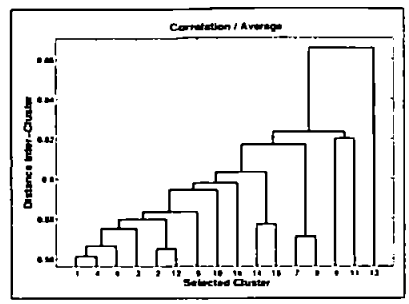
Figure 6.23: Flute - Dendrogram results



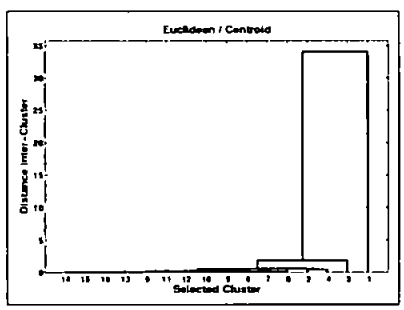
(a) Minkowski / Average



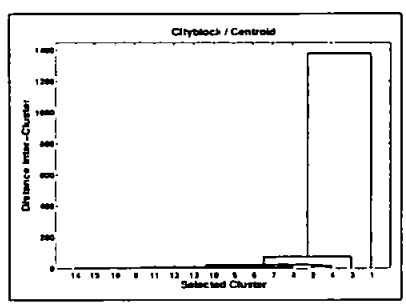
(b) Cosine / Average



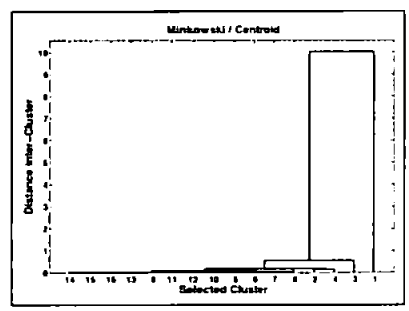
(c) Correlation / Average



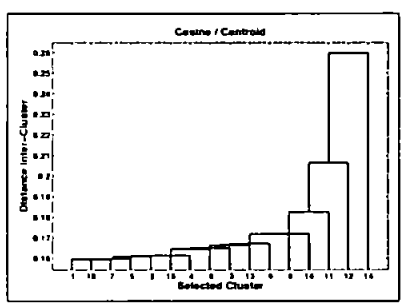
(d) Euclidean / Centroid



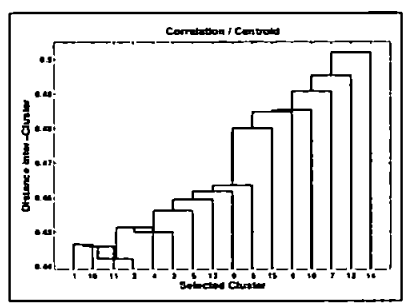
(e) Cityblock / Centroid



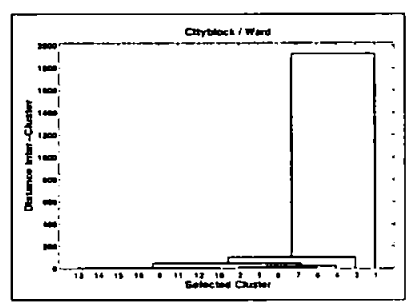
(f) Minkowski / Centroid



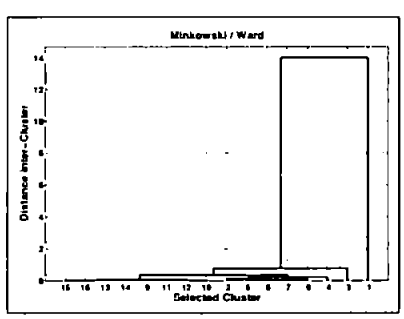
(g) Cosine / Centroid



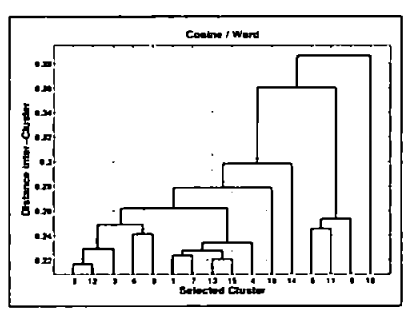
(h) Correlation / Centroid



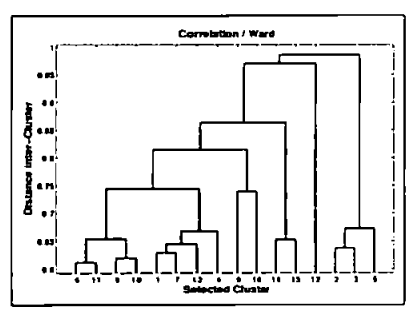
(i) Cityblock / Ward



(j) Minkowski / Ward

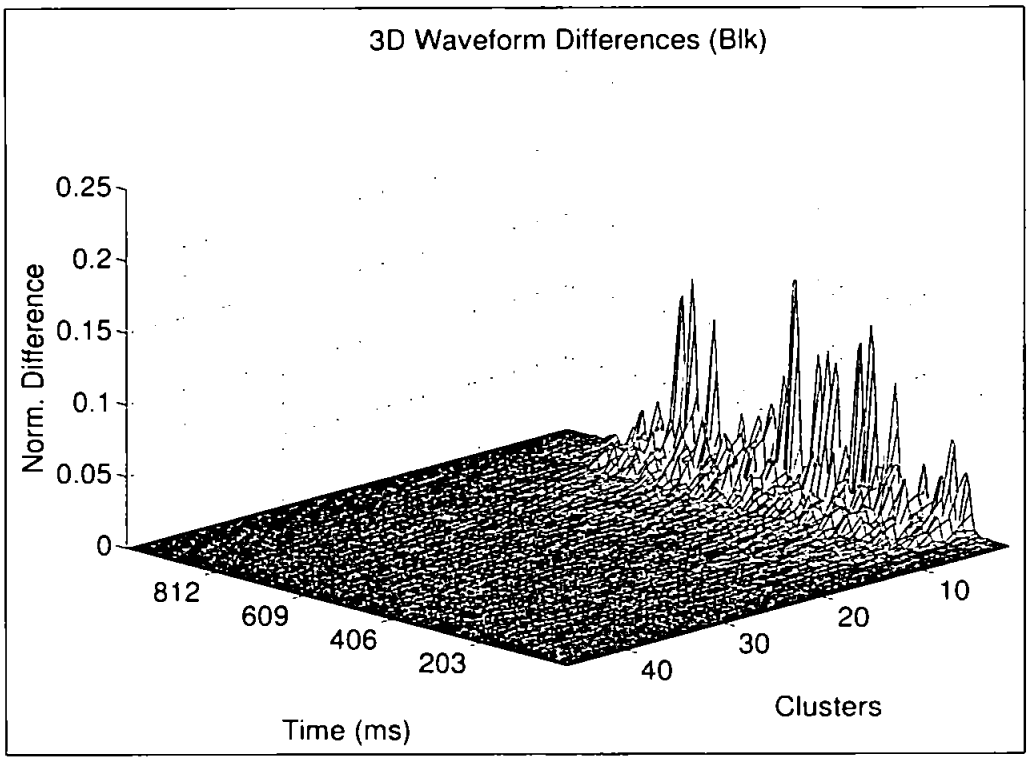


(k) Cosine / Ward

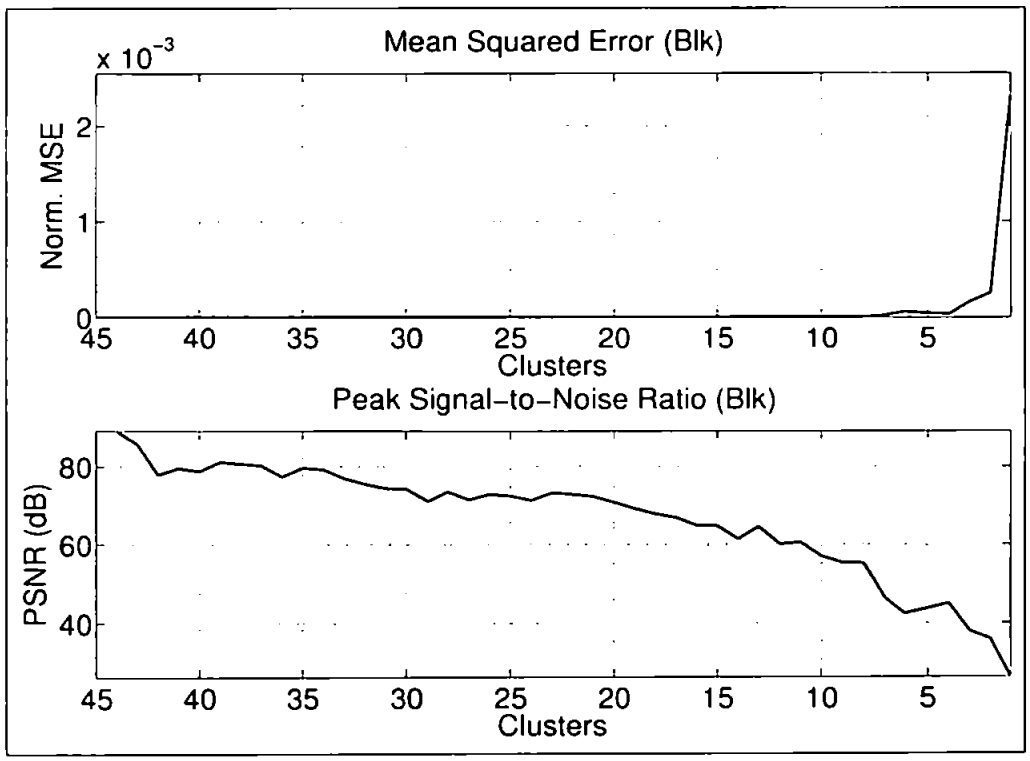


(l) Correlation / Ward

Figure 6.24: Flute - Dendrogram results (*continued*)

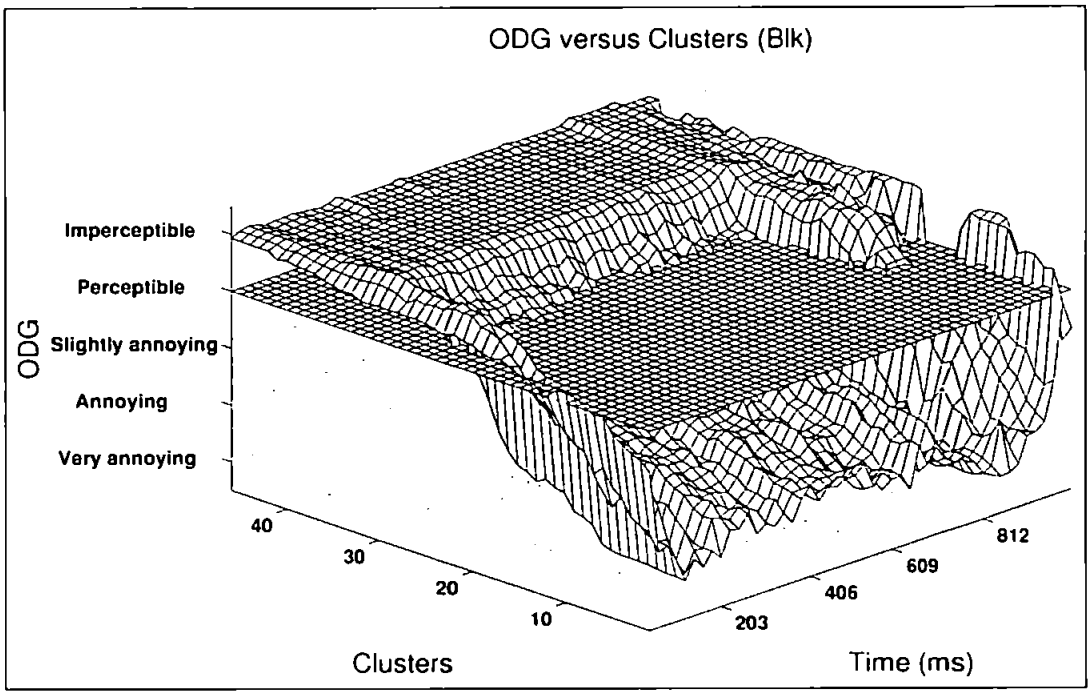


(a) Waveform modeling difference

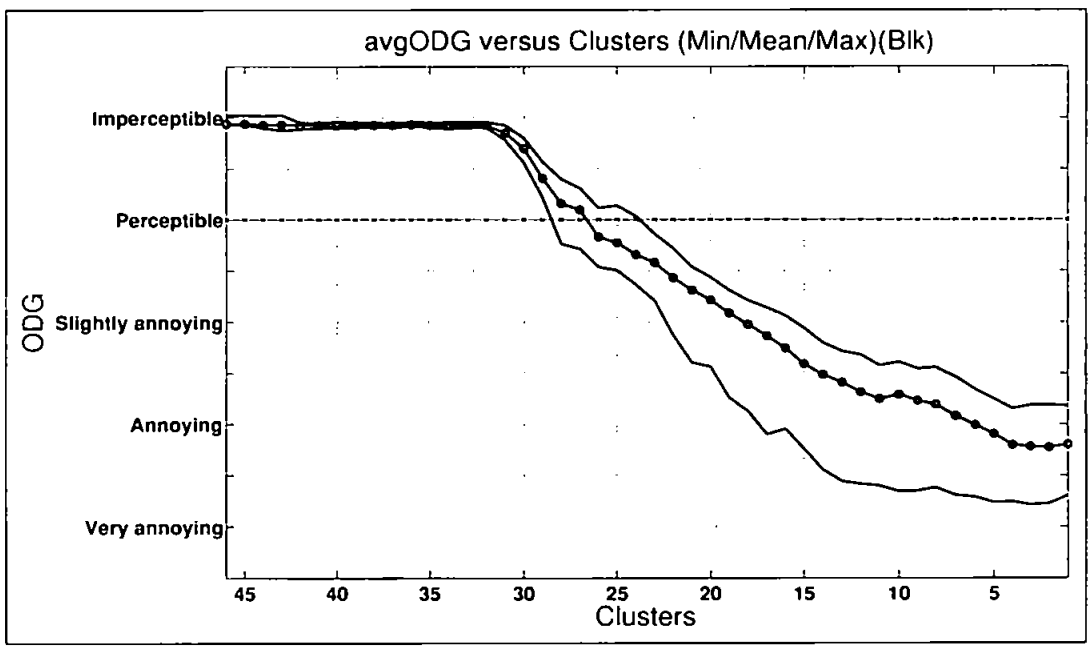


(b) MSE/PSNR Error modeling

Figure 6.25: Flute - Clustering results (Blk)

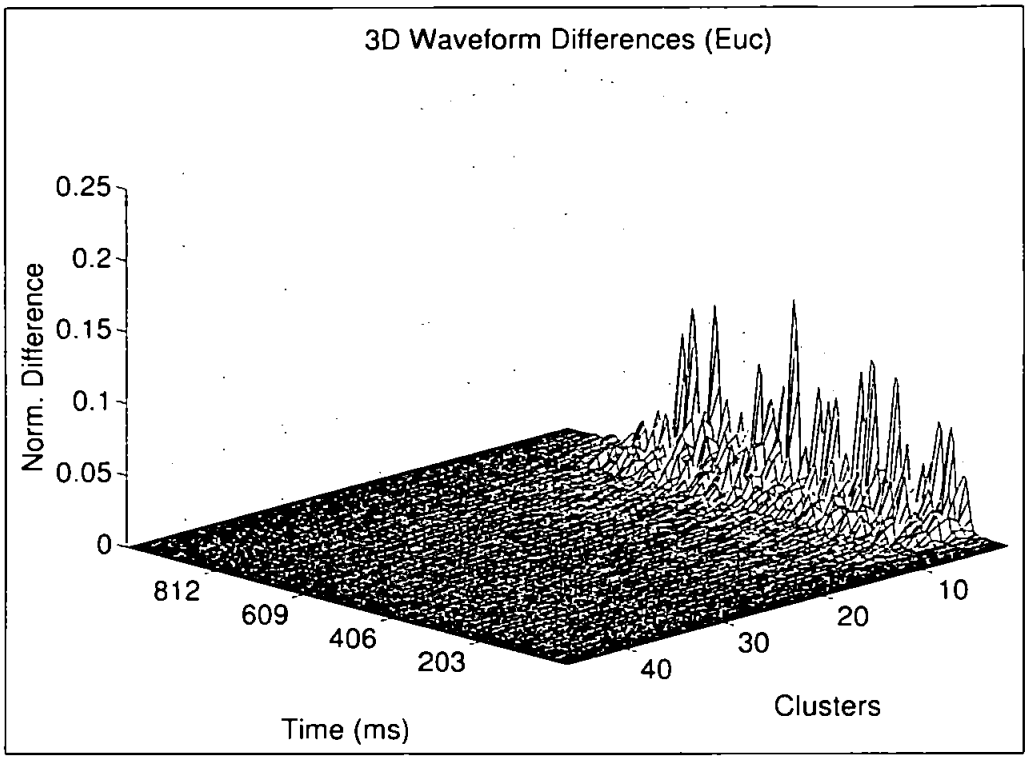


(a) ODGvsCL

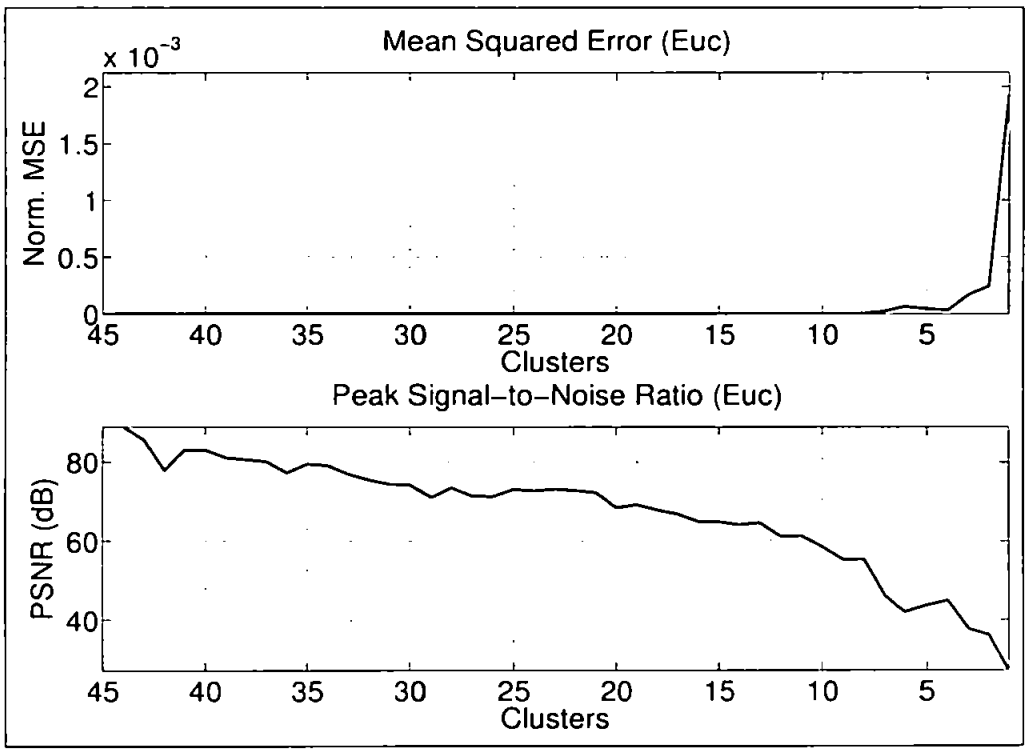


(b) avgODGvsCL

Figure 6.26: Flute - Clustering results ODG(Blk)

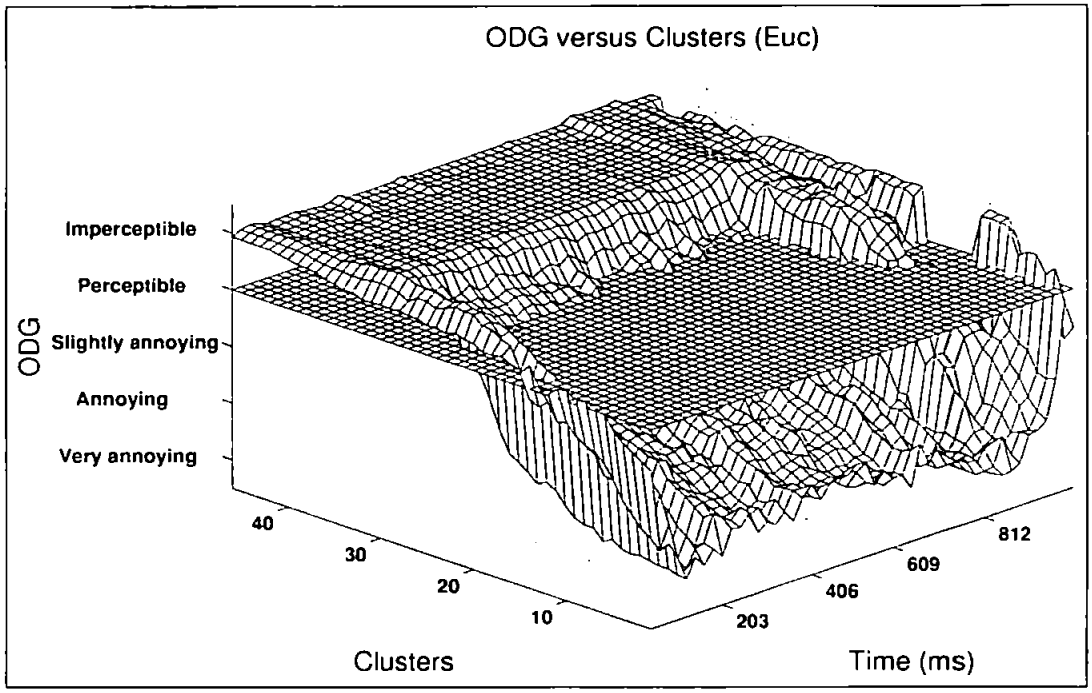


(a) Waveform modeling difference

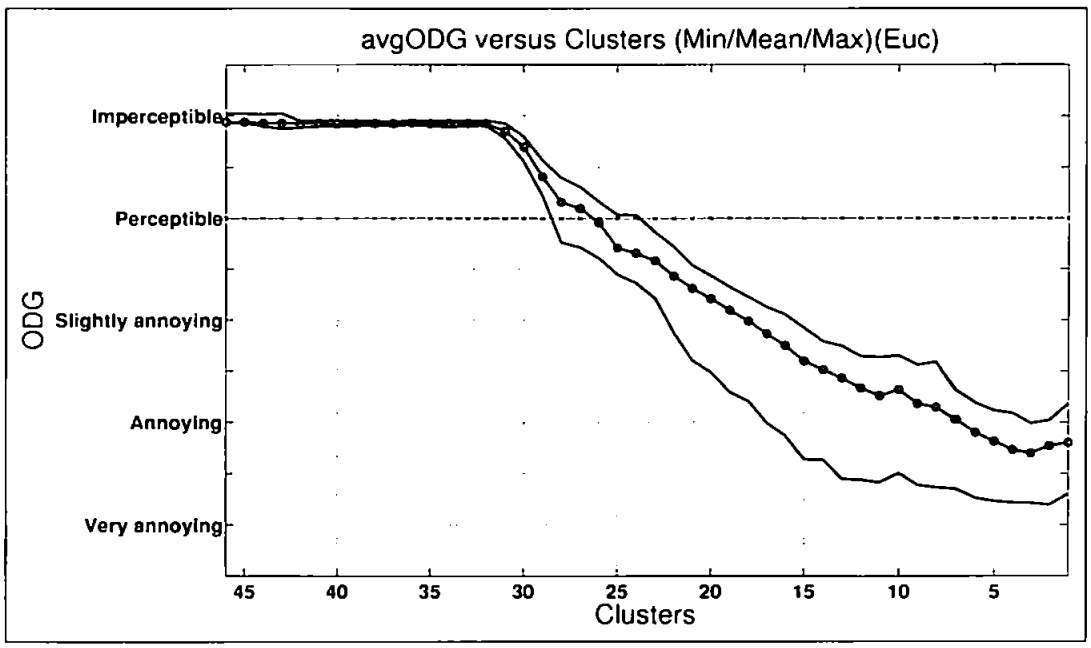


(b) MSE/PSNR Error modeling

Figure 6.27: Flute - Clustering results (Euc)

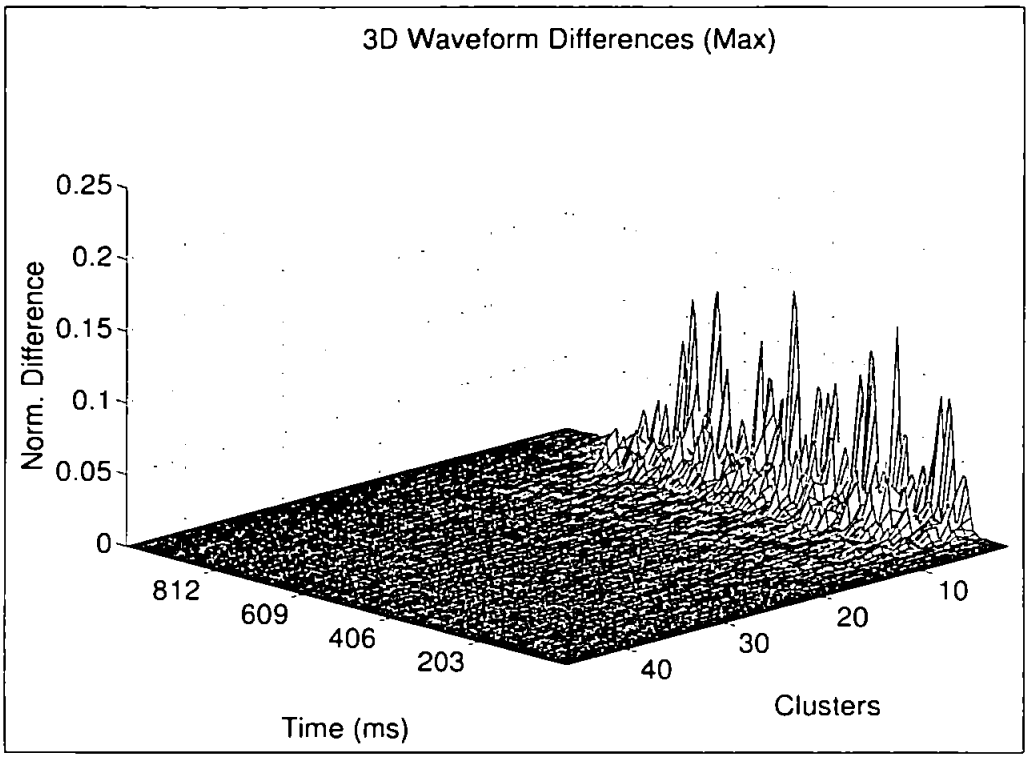


(a) ODGvsCL

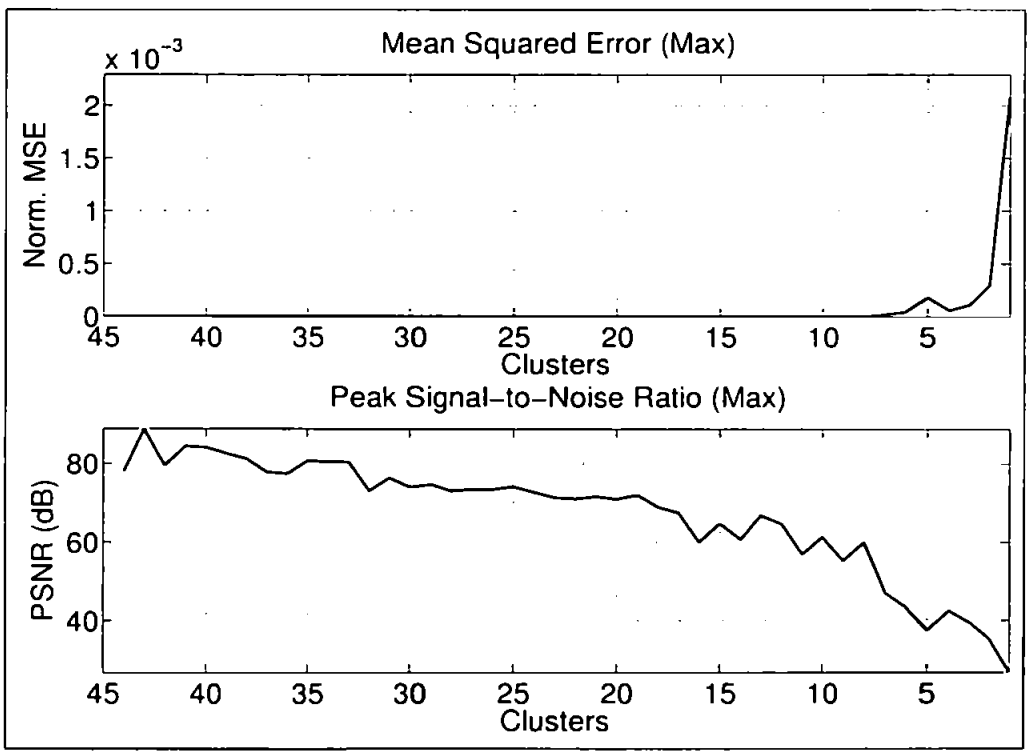


(b) avgODGvsCL

Figure 6.28: Flute - Clustering results ODG (Euc)

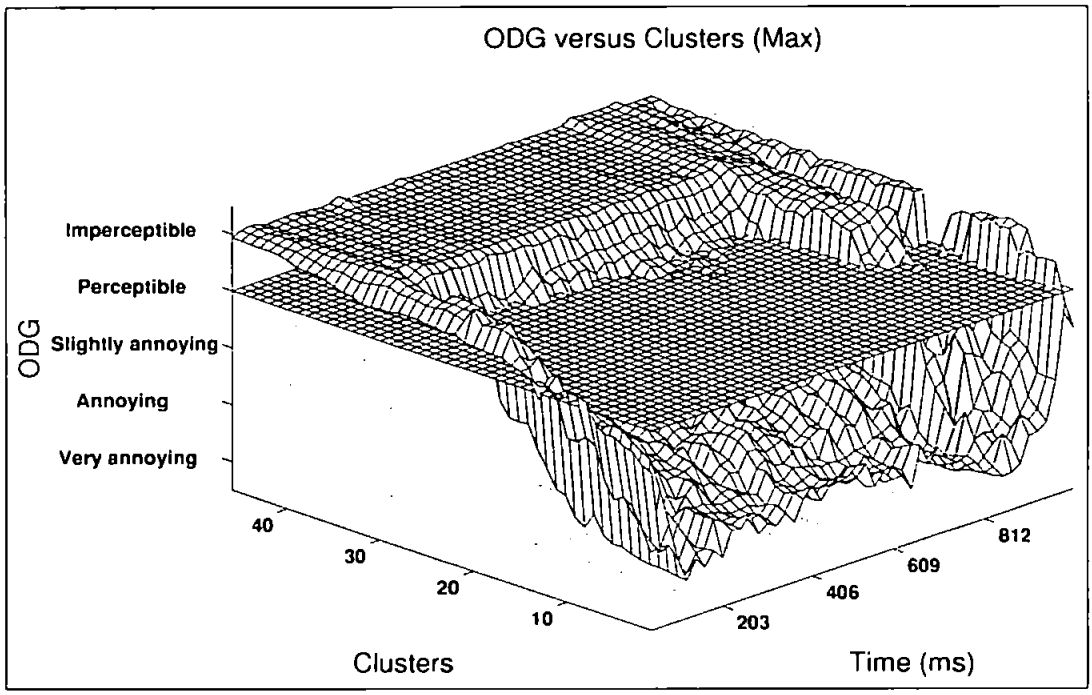


(a) Waveform modeling difference

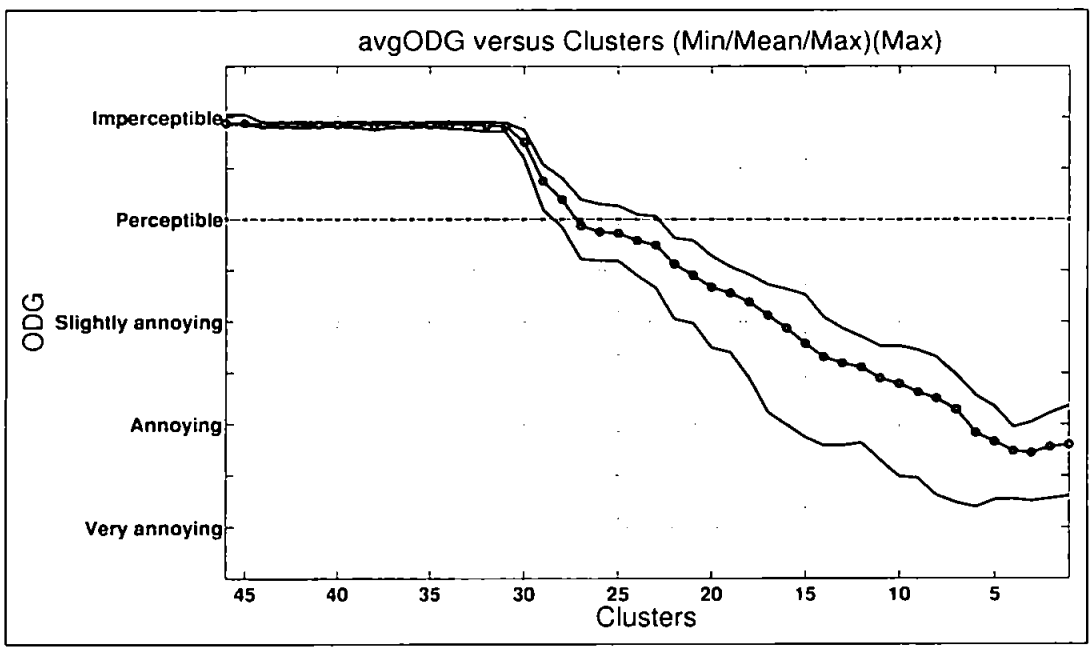


(b) MSE/PSNR Error modeling

Figure 6.29: Flute - Clustering results (Max)

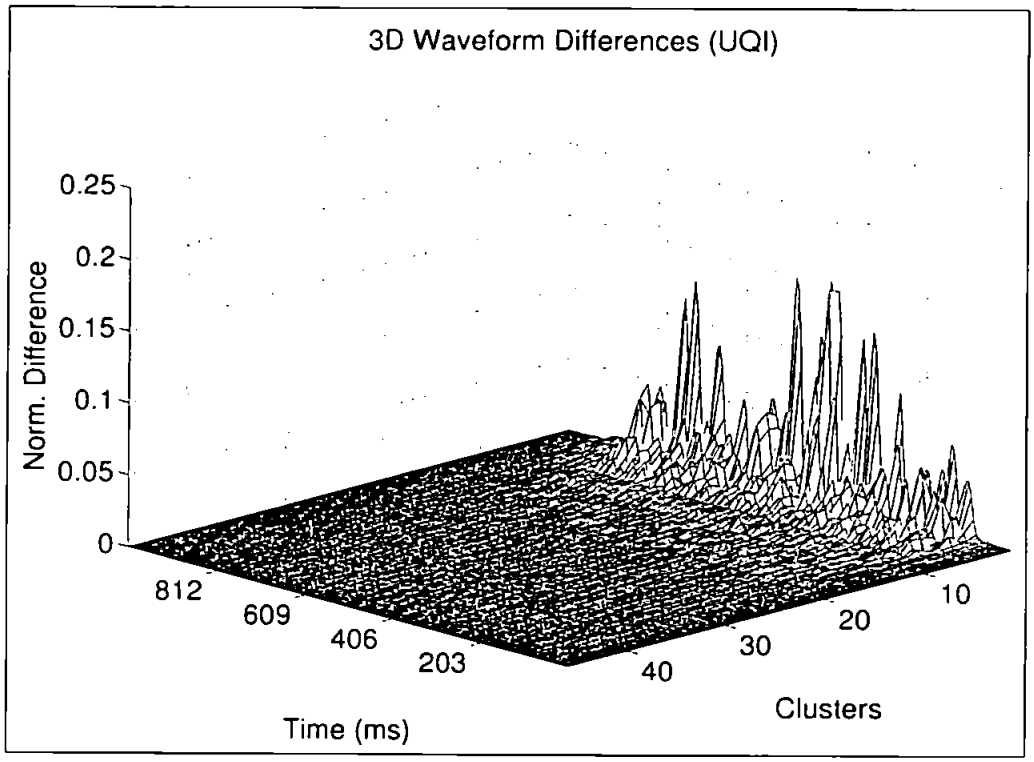


(a) ODGvsCL

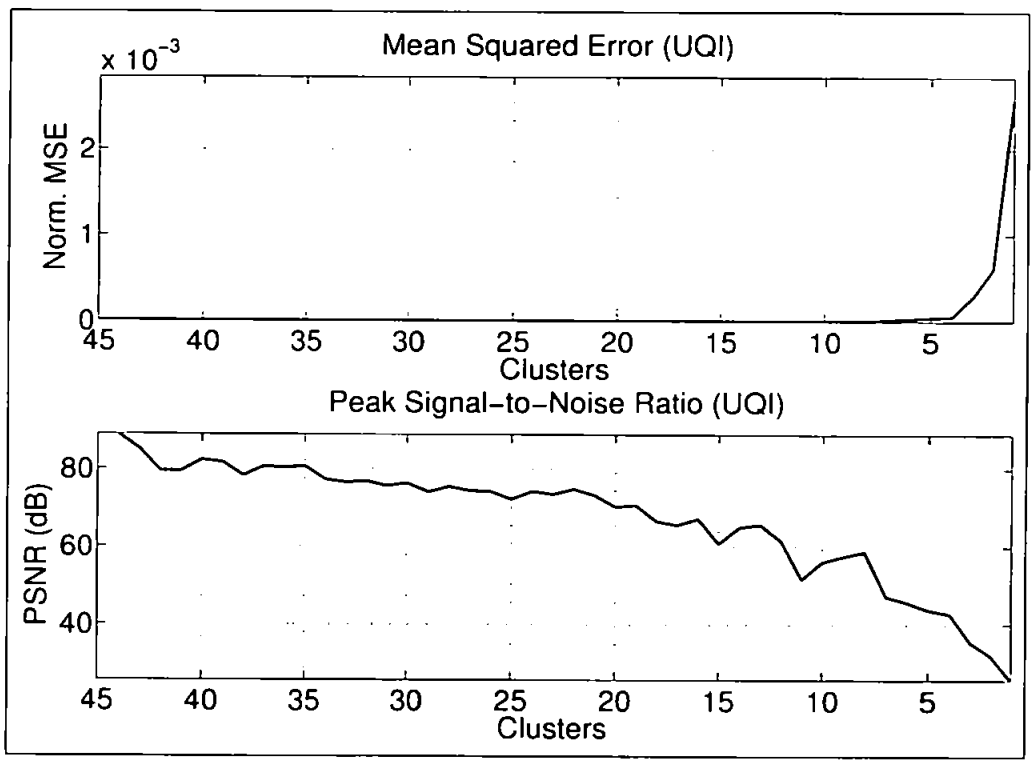


(b) avgODGvsCL

Figure 6.30: Flute - Clustering results ODG (Max)

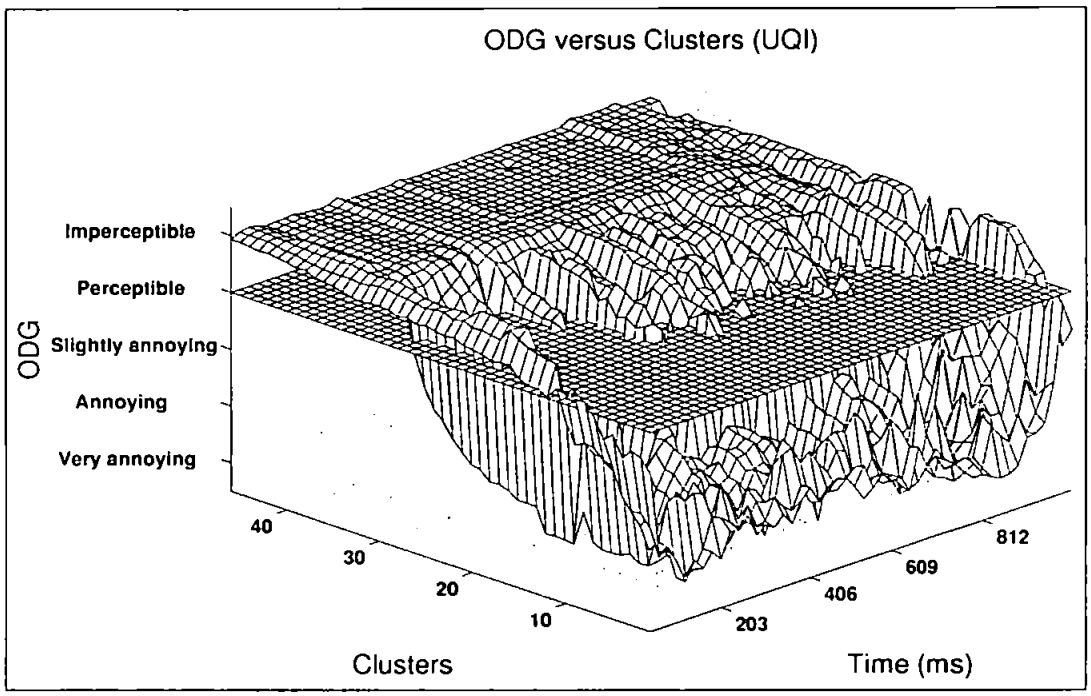


(a) Waveform modeling difference

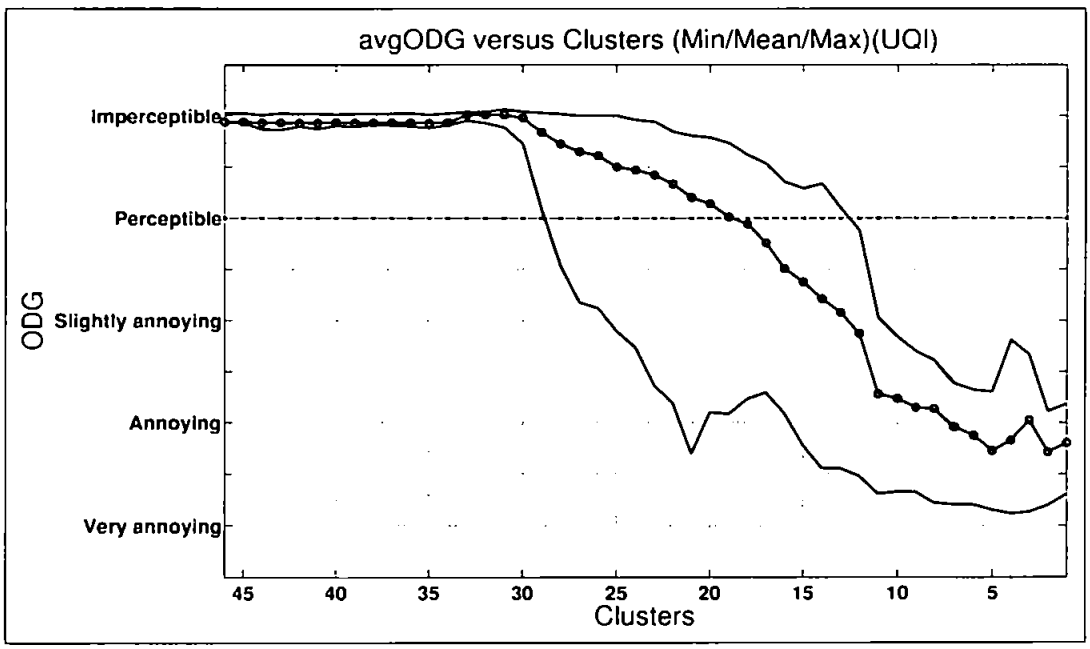


(b) MSE/PSNR Error modeling

Figure 6.31: Flute - Clustering results (UQI)



(a) ODGvsCL



(b) avgODGvsCL

Figure 6.32: Flute - Clustering results ODG (UQI)

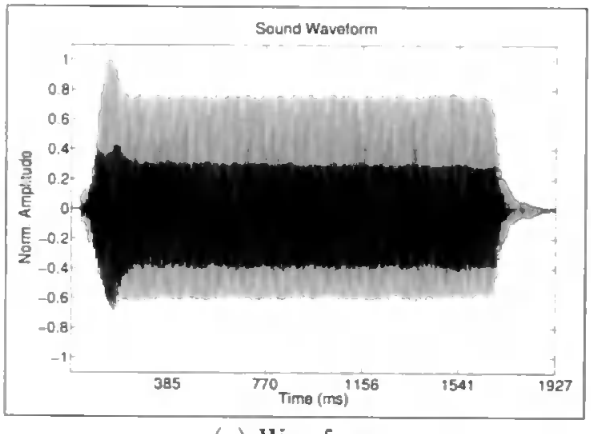
6.4.5 Sound - Horn

Analysis - The sound analysis engine of WinPVoc generated the following results from the modeling of the horn sound. Figure 6.33(a) shows the waveform of this sound, while Figure 6.33(b) show the sound's spectrum in with Figure 6.33(e) and 6.33(f) in spectrum lines (linear and logarithmic scale). Figure 6.33(c) shown that 3 harmonics are overshooting and Figure 6.33(d) allow to investigate this further looking at the 2^{nd}, 3^{rd} and 4^{th} harmonic which have a large maximum to sustain level difference.

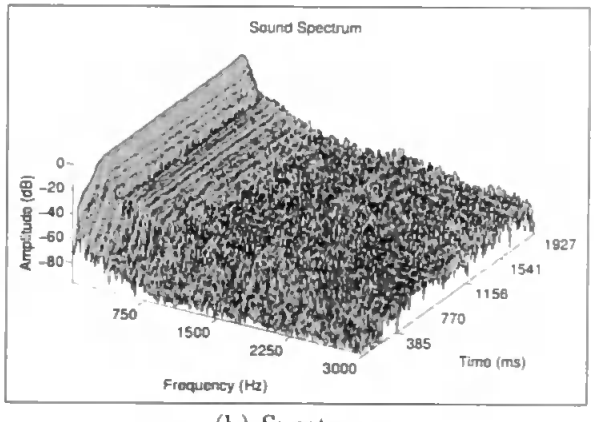
Dendrogram - The AHC clustering algorithm applied to the Horn sound, using different distance metric methods and linkage methods, generated the following results shown in Figures 6.34(a) to 6.34(l) and Figures 6.35(a) to 6.35(l). The results are like all the other cases very different and it is difficult to assess which one should give the best resulting quality, however a majority of them are using the 1^{st} and 2^{nd} harmonic for the last agglomeration which indicate their importance in the final sound character.

Modeling Error - The modeling error results, using 4 distance metric methods (City-block, Euclidean, Max and UQI) are shown in Figure 6.36, 6.38, 6.40 and 6.42. All have a modeling error of inferior to 0.5 and with more error in the final portion of the sound.

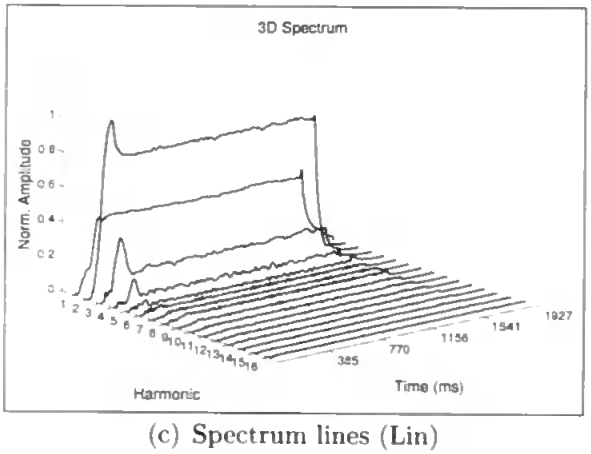
Objective Audio Quality - Figure 6.37, 6.39, 6.41 and 6.43 show the objective audio quality (surfaces and curves) results with Blk, Euc, Max and UQI, methods. The case Max method presents severe errors at the beginning of the clustering process, this seems to be common to sounds with overshooting harmonics in which the specific harmonic is selected at an early stage of the clustering processing, producing some very severe error in the modeling. The UQI method (See Figure 6.43(b)) provides a much more gradual modeling errors compared to the other cases.



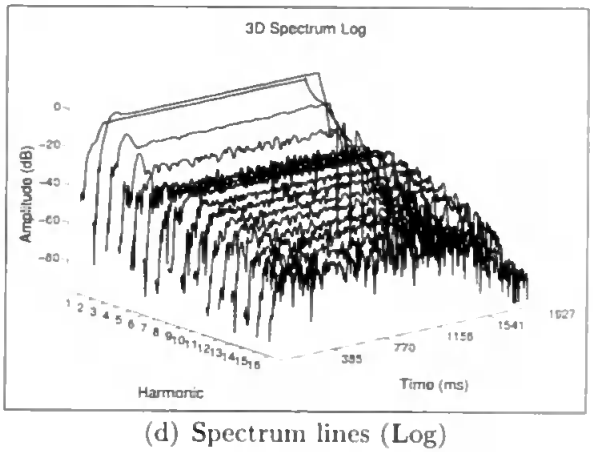
(a) Waveform



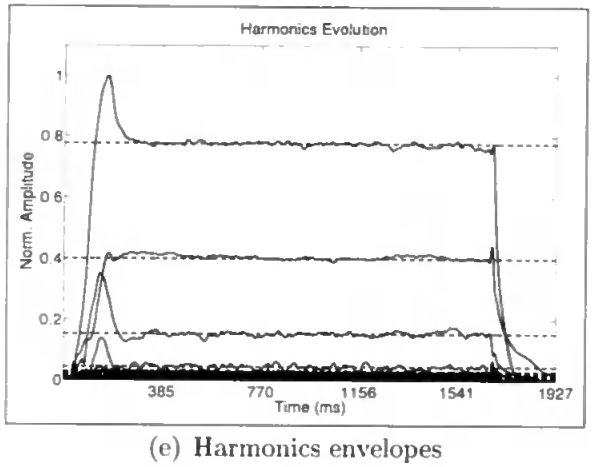
(b) Spectrum



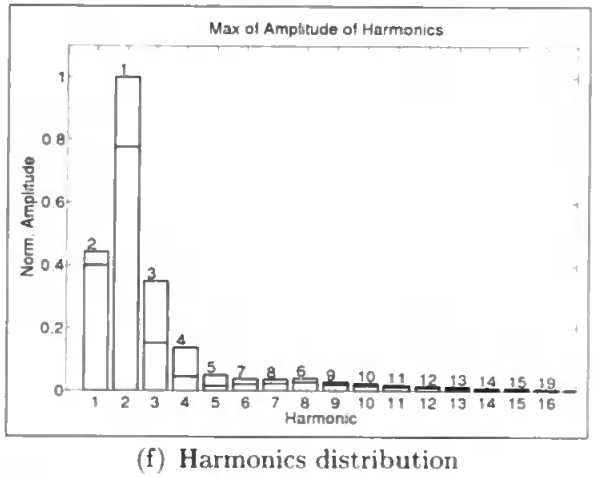
(c) Spectrum lines (Lin)



(d) Spectrum lines (Log)

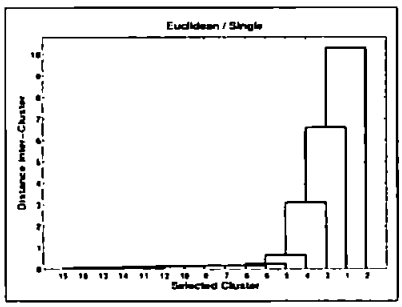


(e) Harmonics envelopes

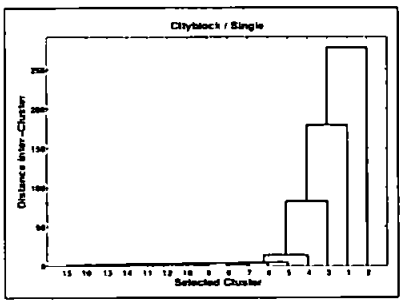


(f) Harmonics distribution

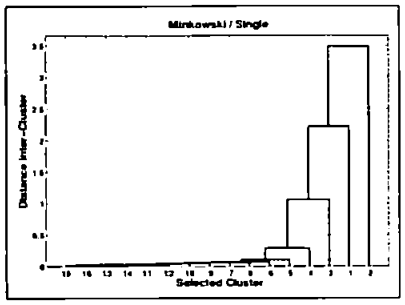
Figure 6.33: Horn - Sound analysis



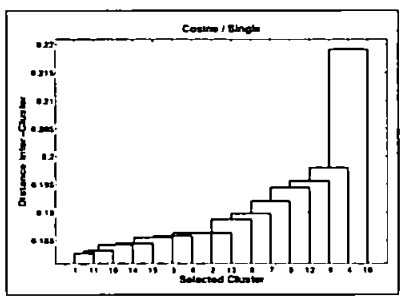
(a) Euclidean / Single



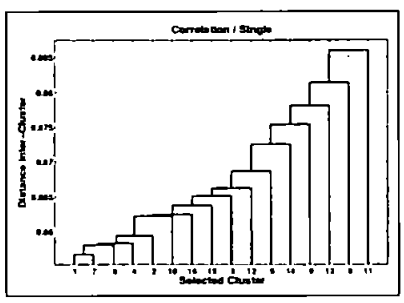
(b) Cityblock / Single



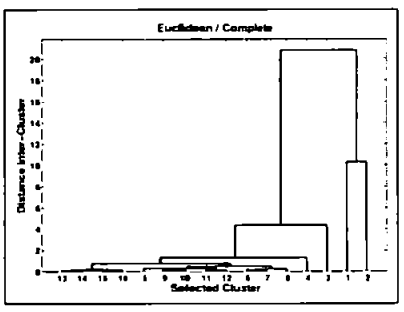
(c) Minkowski / Single



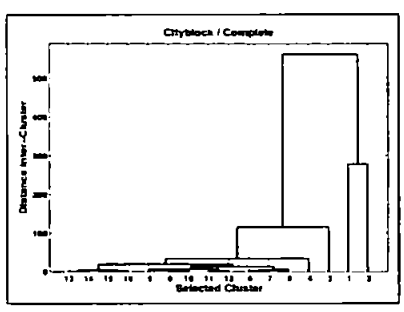
(d) Cosine / Single



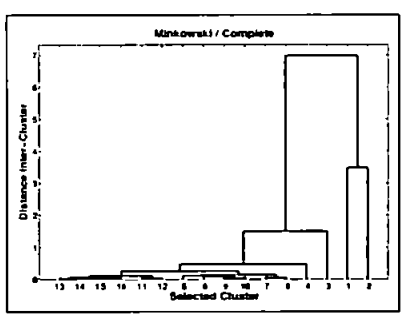
(e) Correlation / Single



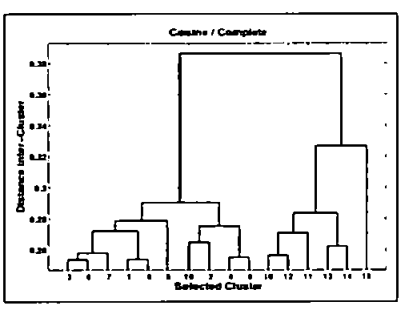
(f) Euclidean / Complete



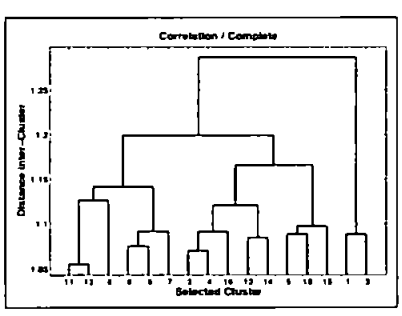
(g) Cityblock / Complete



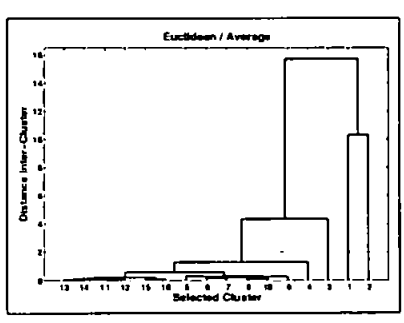
(h) Minkowski / Complete



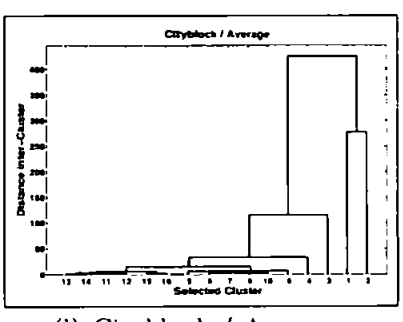
(i) Cosine / Complete



(j) Correlation / Complete

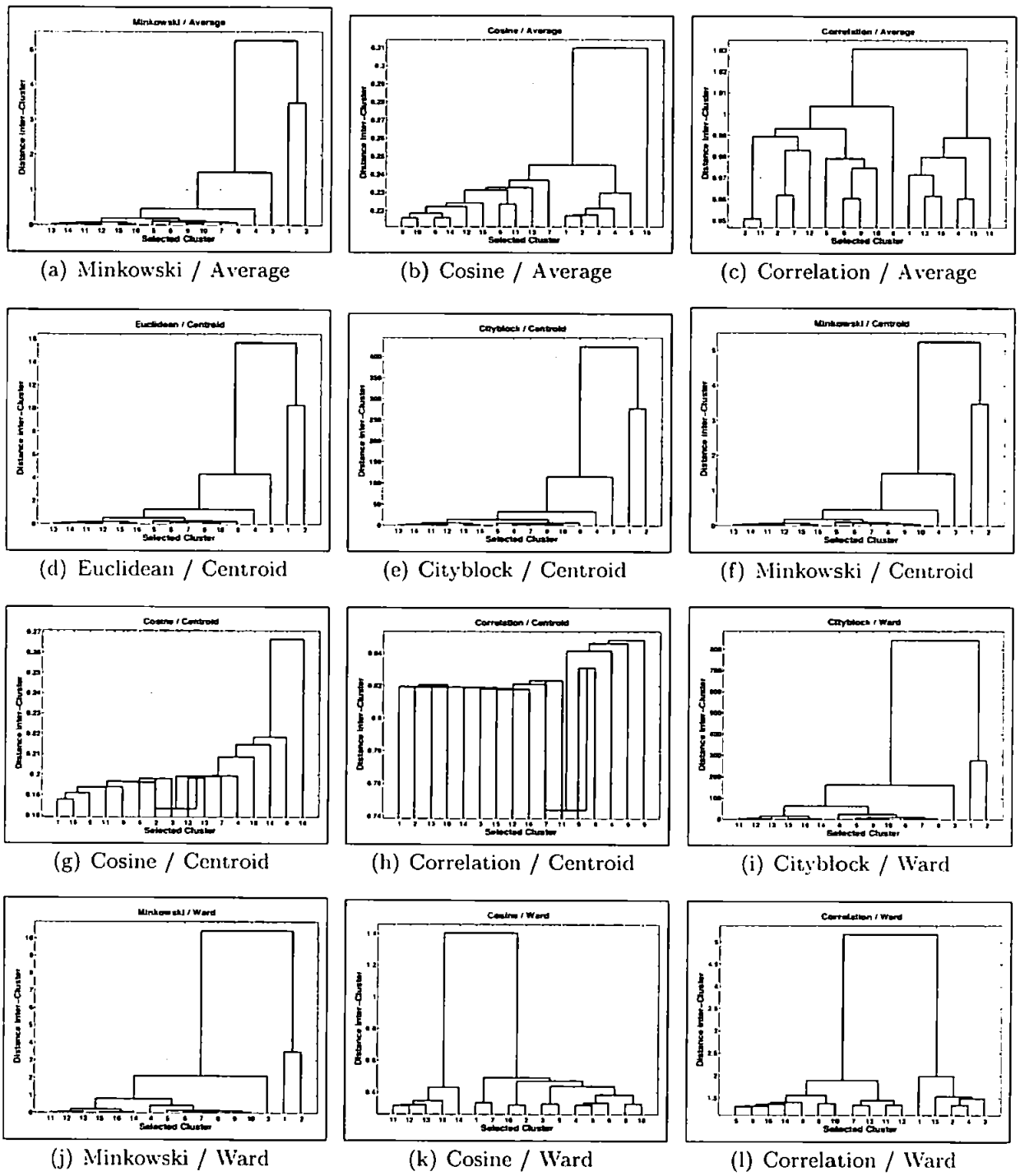


(k) Euclidean / Average



(l) Cityblock / Average

Figure 6.34: Horn - Dendrogram results



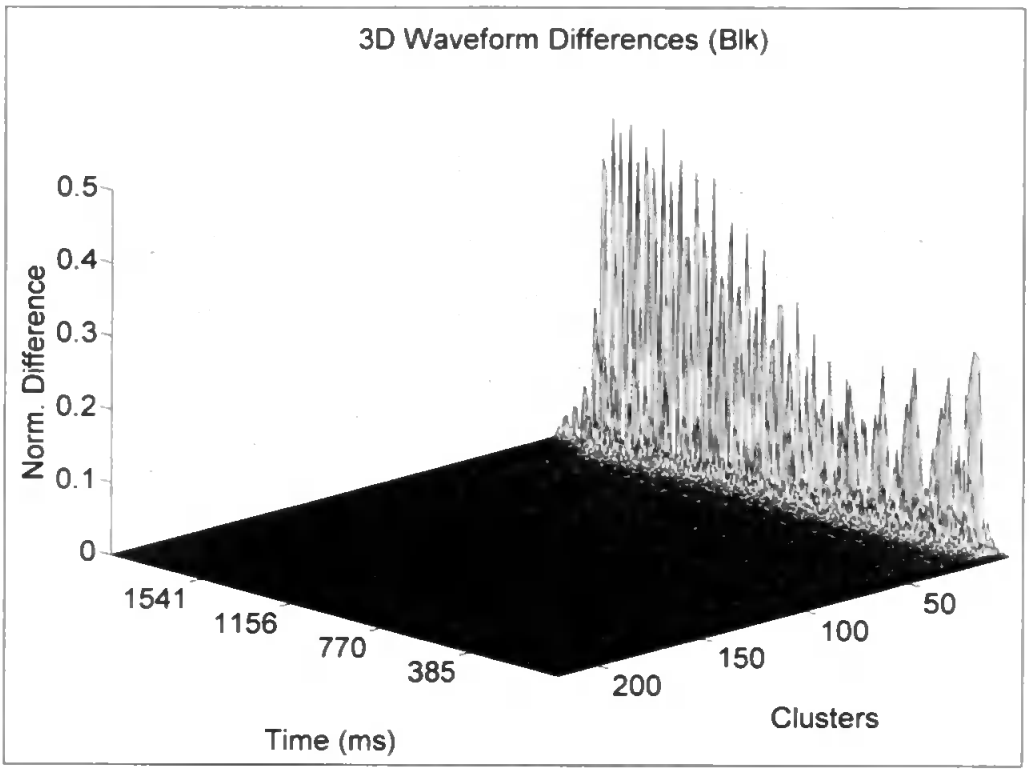
(a) Minkowski / Average (b) Cosine / Average (c) Correlation / Average

(d) Euclidean / Centroid (e) Cityblock / Centroid (f) Minkowski / Centroid

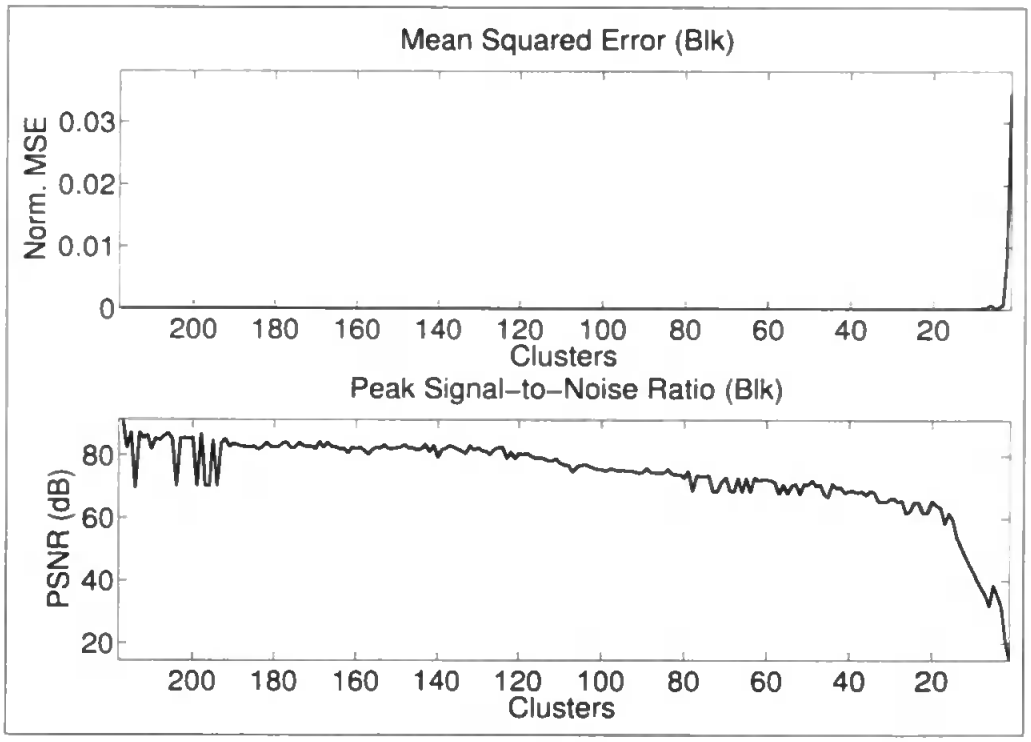
(g) Cosine / Centroid (h) Correlation / Centroid (i) Cityblock / Ward

(j) Minkowski / Ward (k) Cosine / Ward (l) Correlation / Ward

Figure 6.35: Horn - Dendrogram results (*continued*)

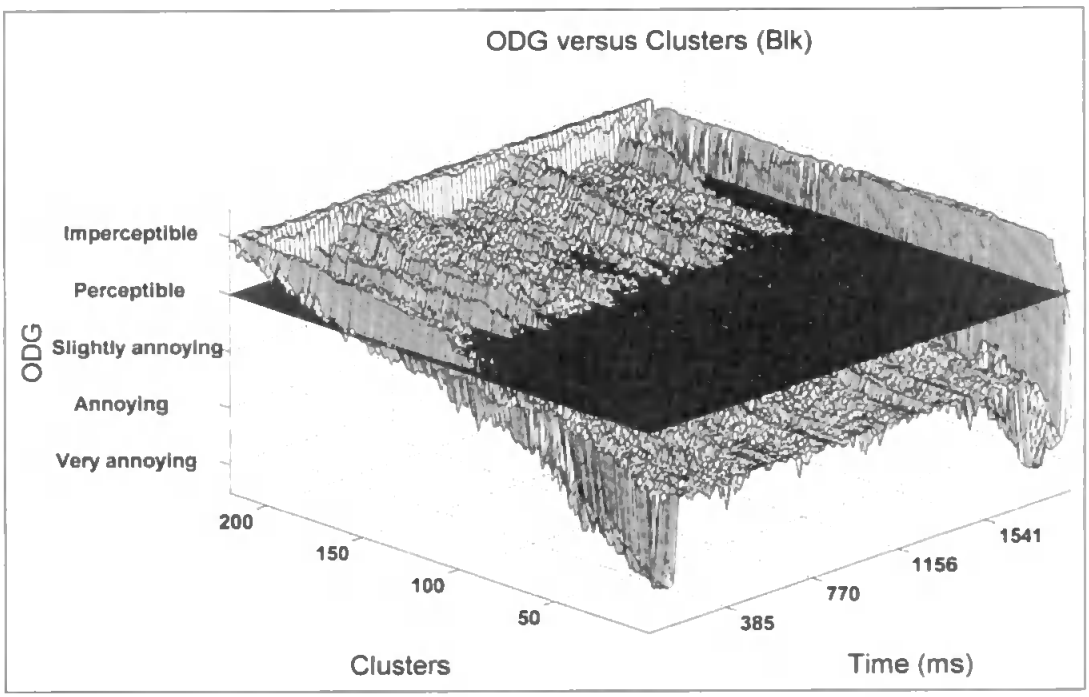


(a) Waveform modeling difference

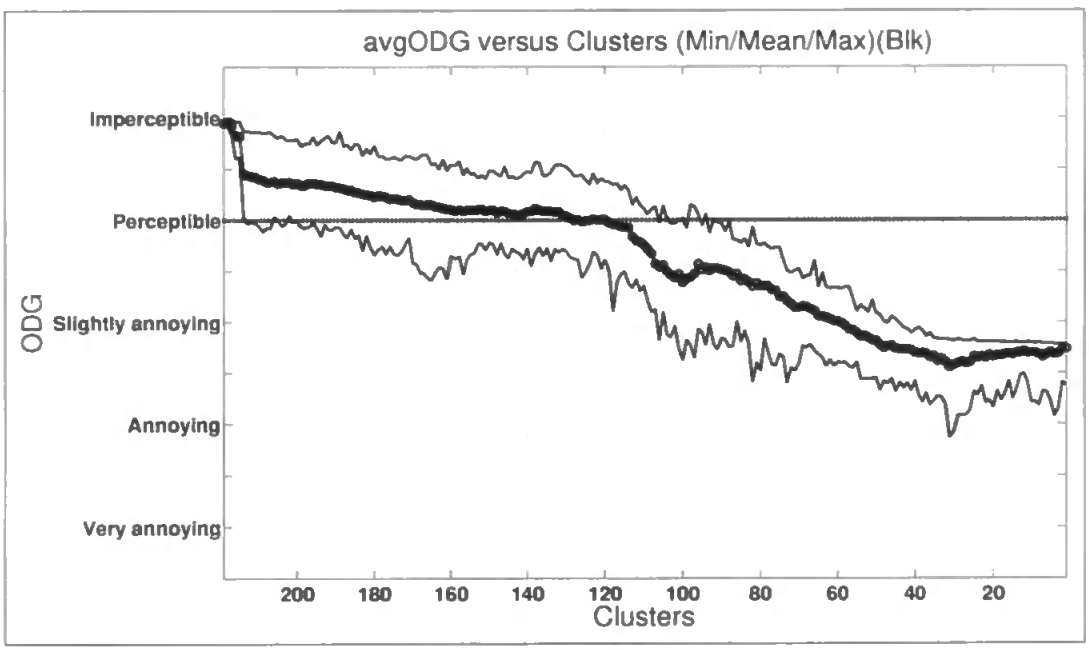


(b) MSE/PSNR Error modeling

Figure 6.36: Horn - Clustering results (Blk)

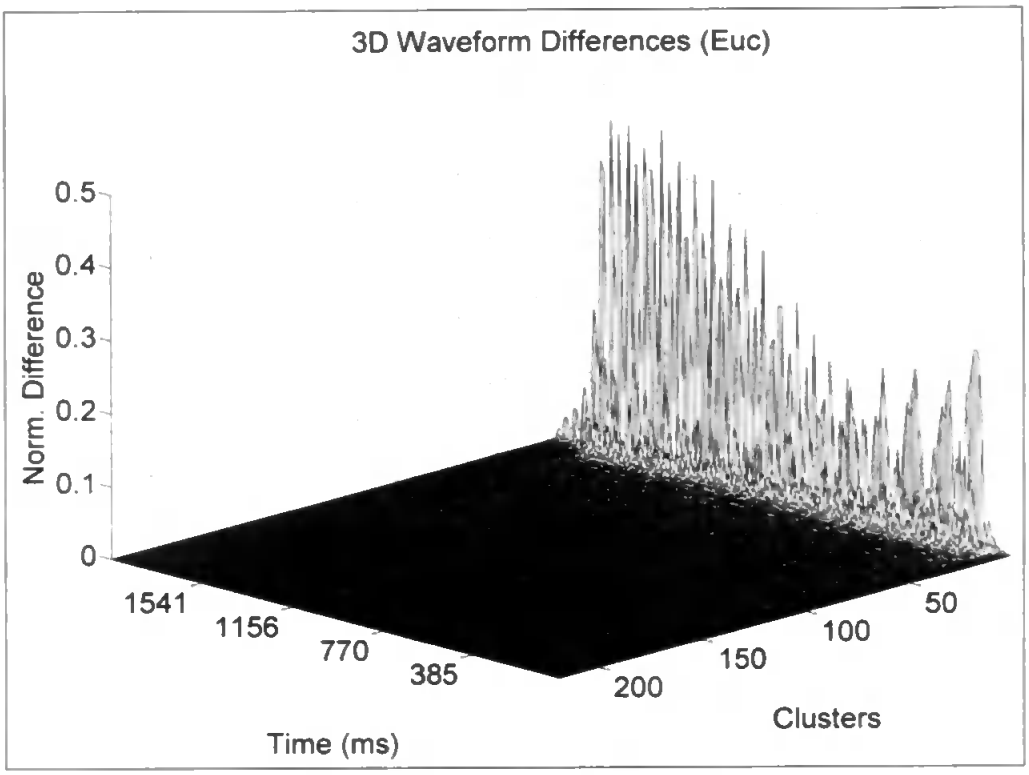


(a) ODGvsCL

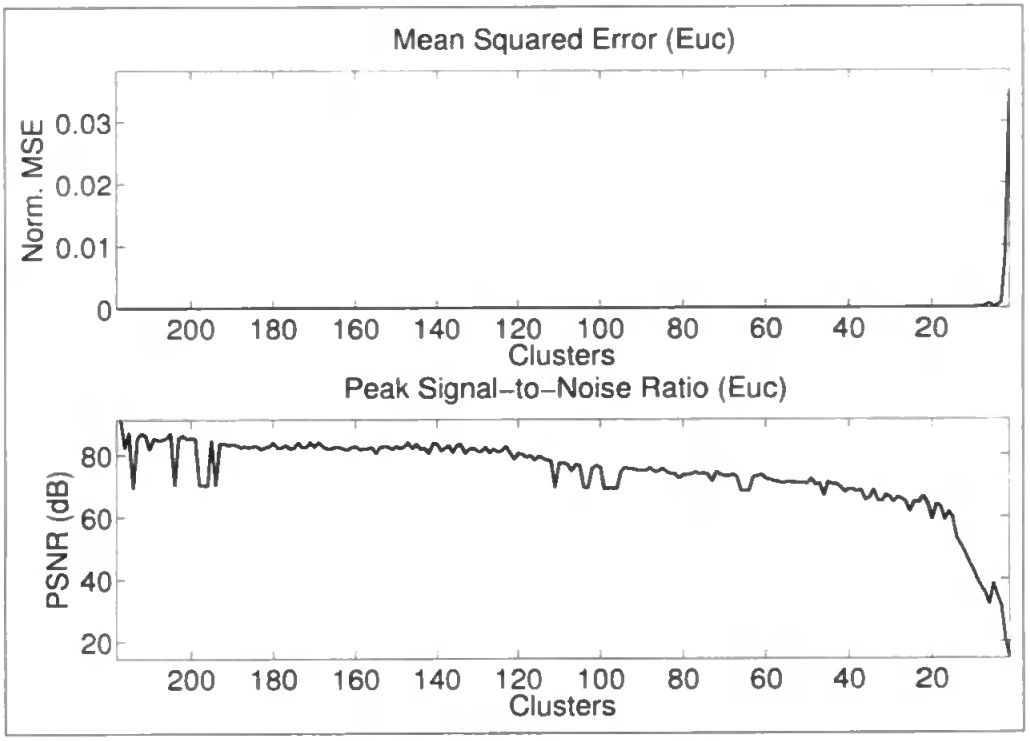


(b) avgODGvsCL

Figure 6.37: Horn - Clustering results ODG(Blk)

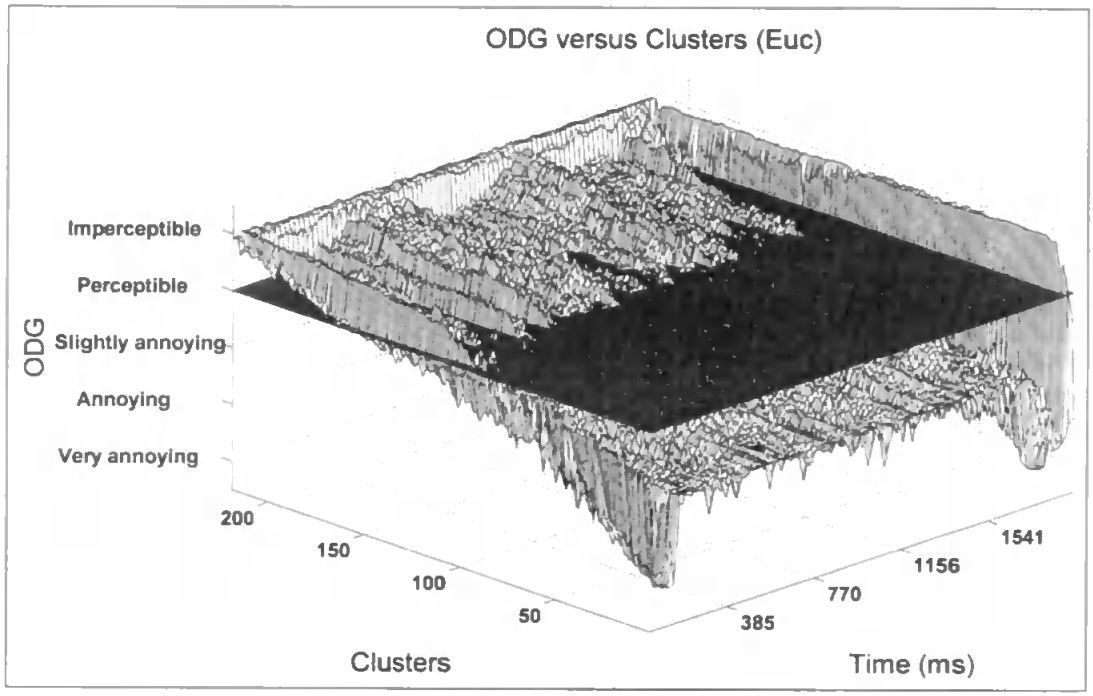


(a) Waveform modeling difference

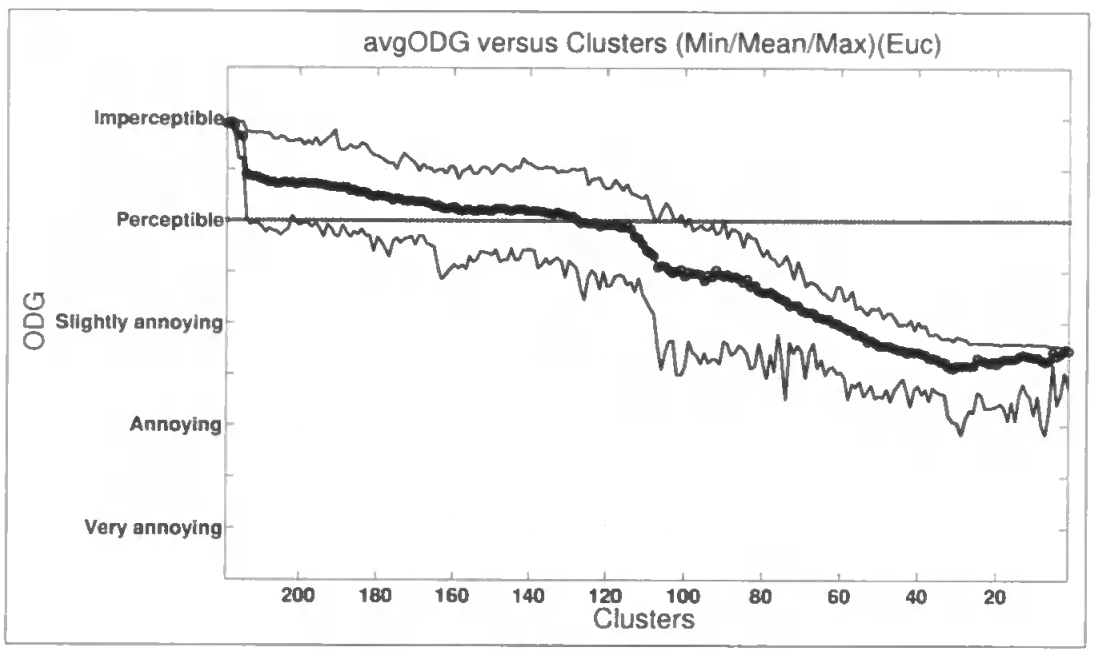


(b) MSE/PSNR Error modeling

Figure 6.38: Horn - Clustering results (Euc)

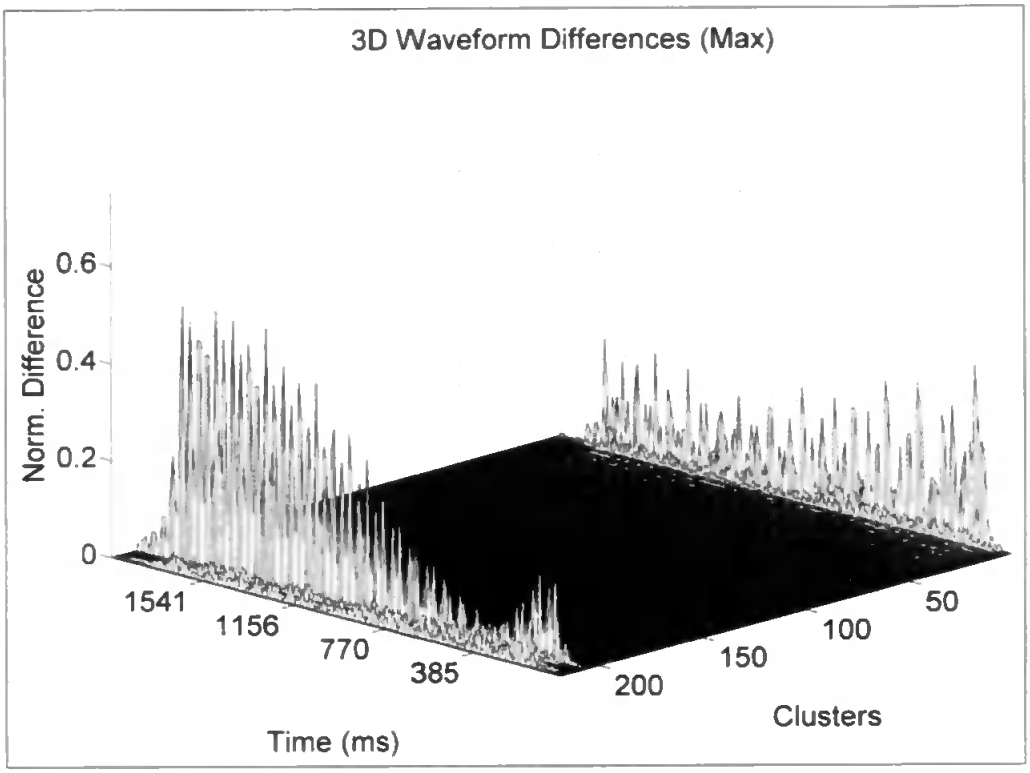


(a) ODGvsCL

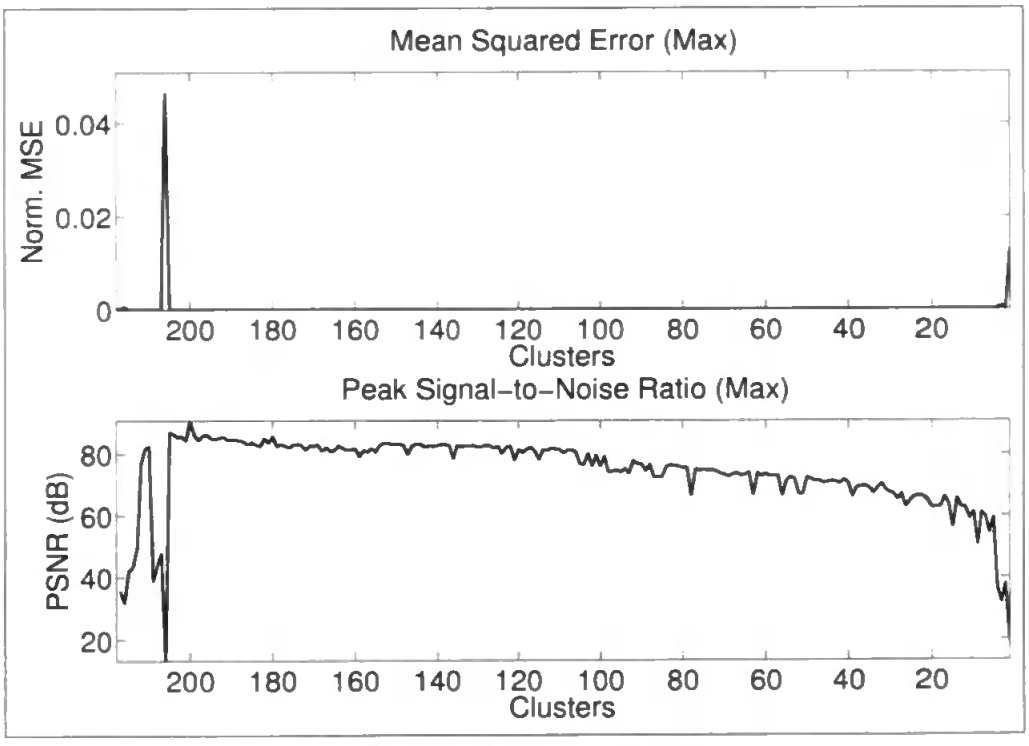


(b) avgODGvsCL

Figure 6.39: Horn - Clustering results ODG (Euc)

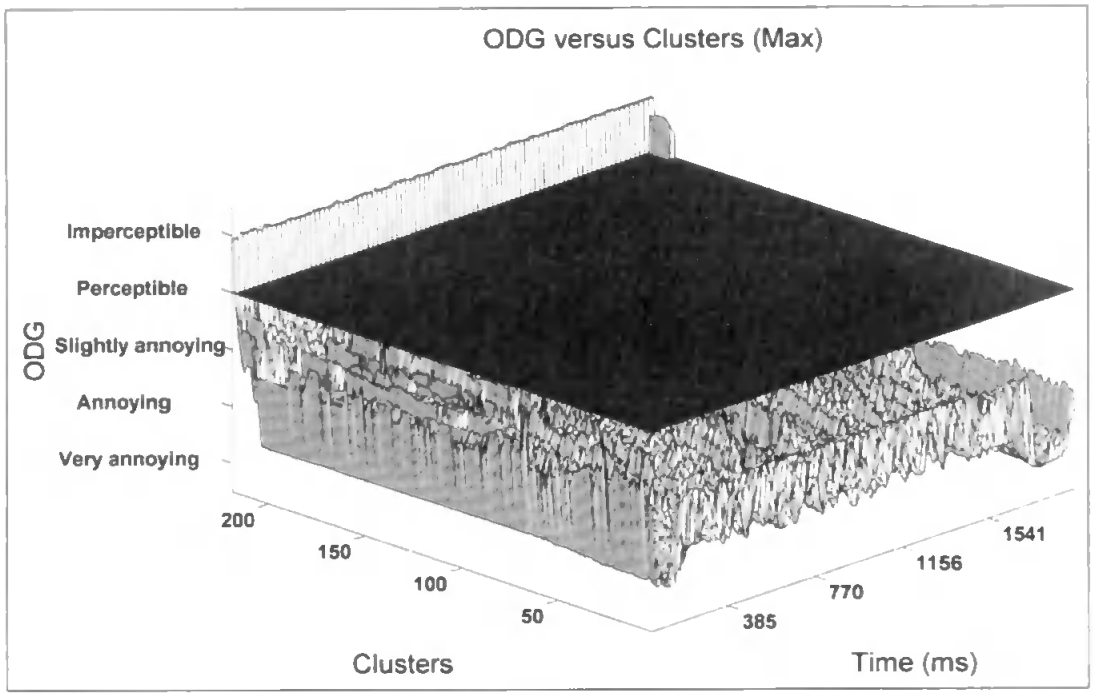


(a) Waveform modeling difference

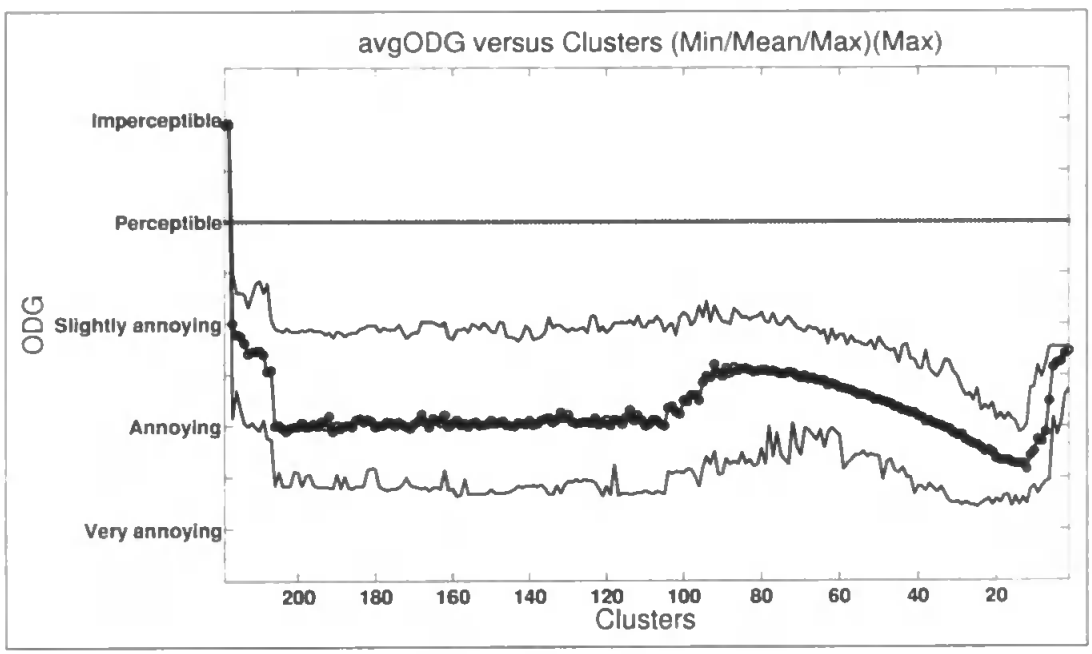


(b) MSE/PSNR Error modeling

Figure 6.40: Horn - Clustering results (Max)

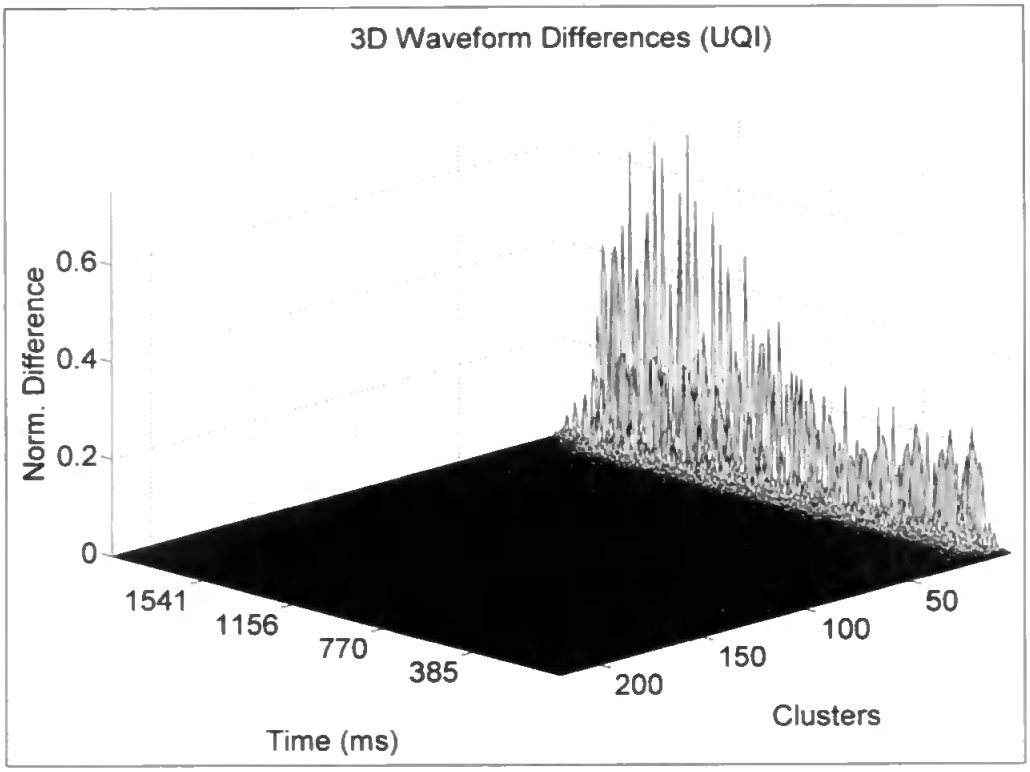


(a) ODGvsCL

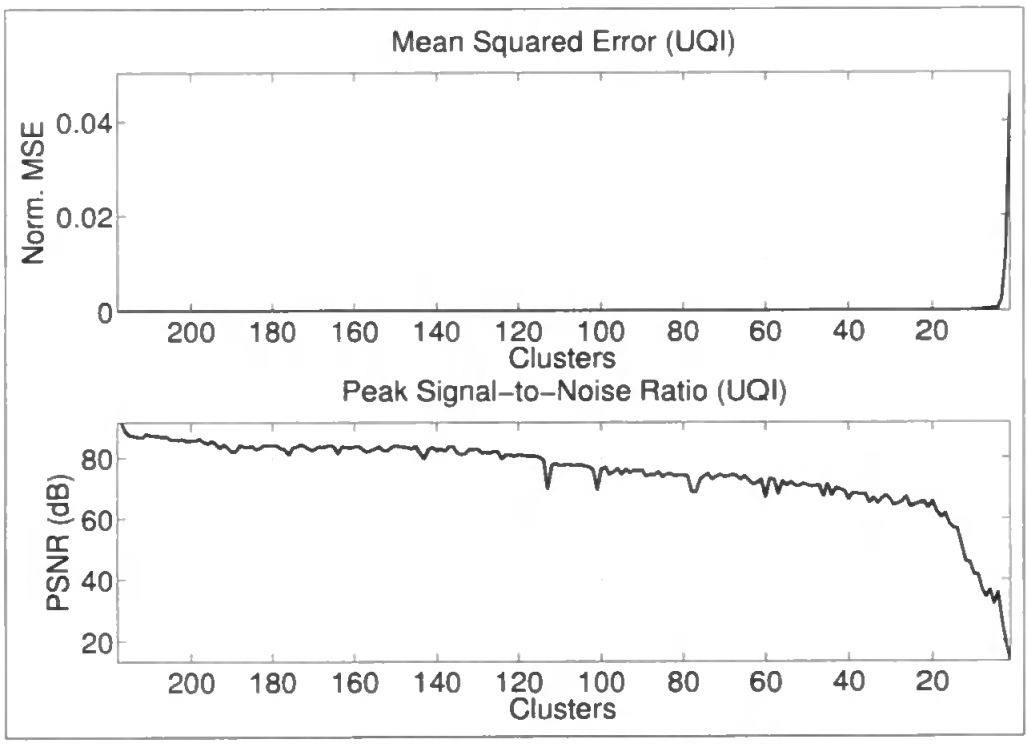


(b) avgODGvsCL

Figure 6.41: Horn - Clustering results ODG (Max)

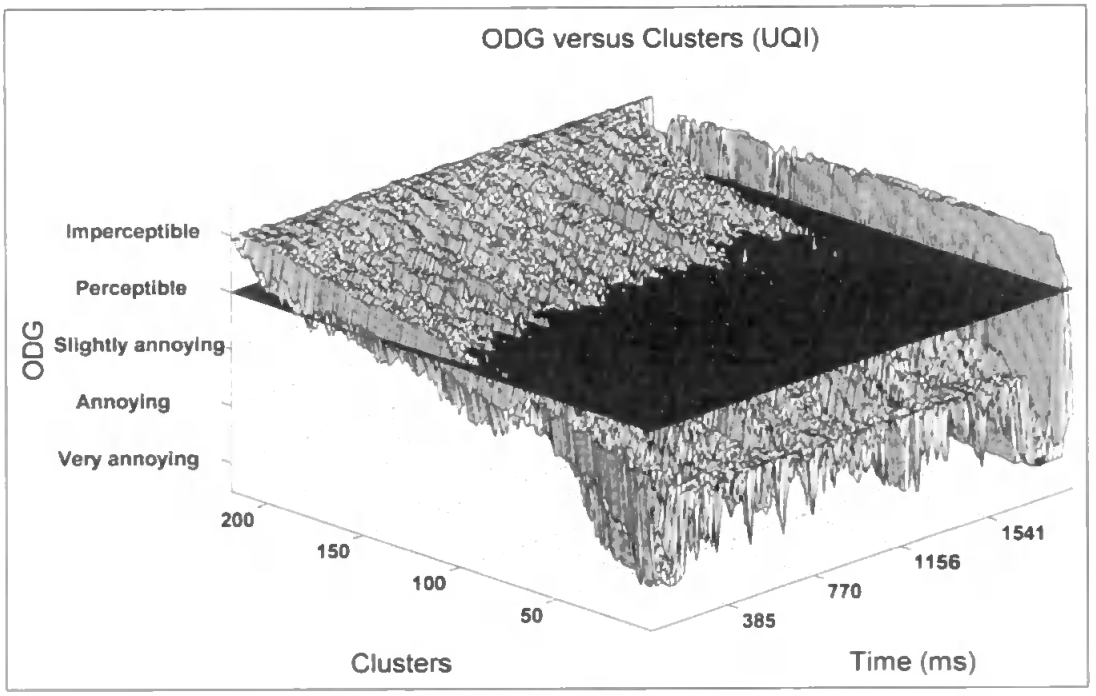


(a) Waveform modeling difference

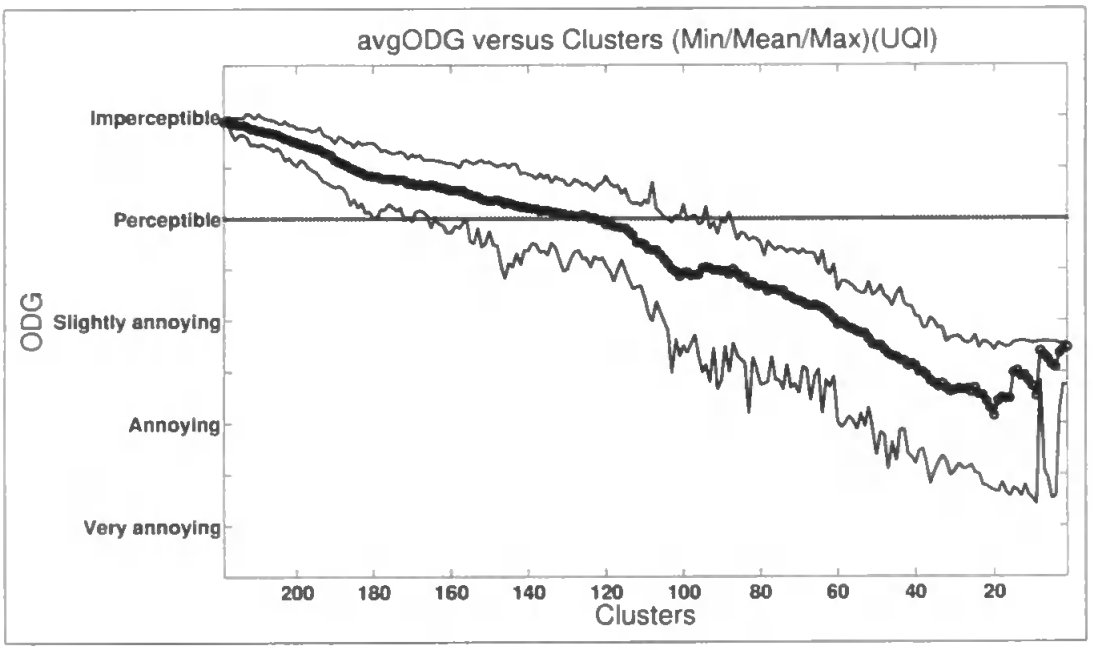


(b) MSE/PSNR Error modeling

Figure 6.42: Horn - Clustering results (UQI)



(a) ODGvsCL



(b) avgODGvsCL

Figure 6.43: Horn - Clustering results ODG (UQI)

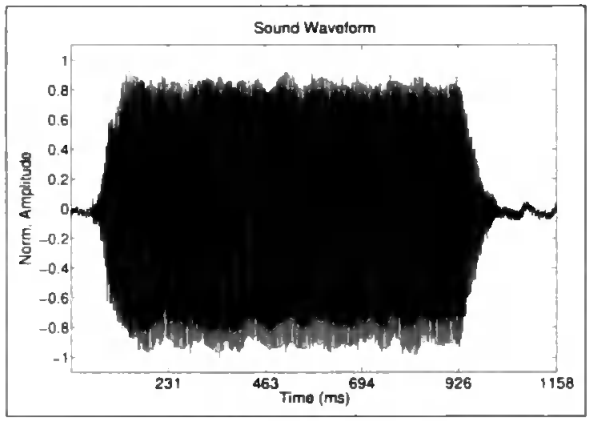
6.4.6 Sound - Principal 4'

Analysis - Figure 6.44(a) shows the waveform of the Principal 4' sound, with Figure 6.44(b) showing its spectrum. Figure 6.44(c) and Figure 6.44(d) show the spectrum lines version in linear and logarithmic scale respectively. The harmonics envelopes and distribution are shown in Figure 6.44(e) and Figure 6.44(f). Two harmonic seems to be predominant, the 1^{st} and the 3^{rd} with a lot of amplitude modulation. The 5^{th} harmonic presents a large overshooting behavior.

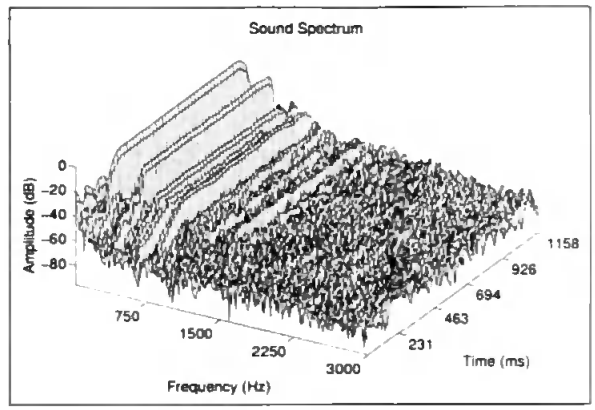
Dendrogram - Figures 6.45(a) to 6.45(l) and Figures 6.46(a) to 6.46(l) show many results using *single* linkage. A majority of them are using the 1^{st} and 3^{nd} harmonic for the last agglomeration indicating their importance in the final sound character. Correlation and cosine methods always provide very *spread* shaped dendrogram.

Modeling Error - Figure 6.47, Figure 6.49, Figure 6.51 and Figure 6.53 show the modeling errors results of the AHC algorithm using 4 distance metric methods (Cityblock, Euclidean, Max and UQI). All 4 present a relative high error with a maximum below the 0.7 value.

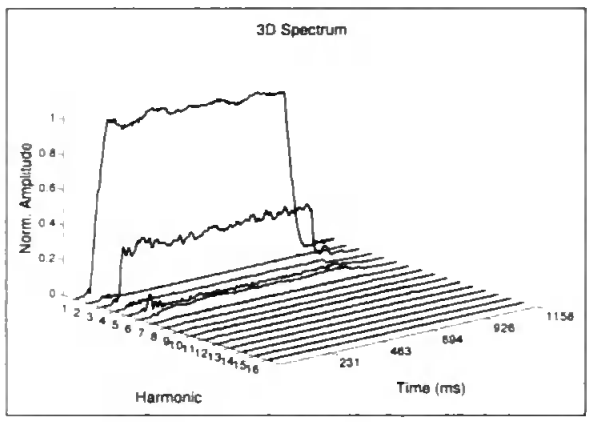
Objective Audio Quality - Figures 6.48, 6.50, 6.52 and 6.54(b) show the objective audio quality surfaces for the Principal 4' sound with the AHC algorithm and the Cityblock, Euclidean, Max and UQI distance metric methods. Similar to all the other objective audio quality surfaces, an additional "PEAQ Perceptible plan" (i.e. ODG = -1) is shown to help visual inspection. As the number of cluster decrease, the surface falls and cut the PEAQ Perceptible plan indicating some perceptible audio degradation in the synthesised sound. The Cityblock and Euclidean cases shown that decreasing the clusters down to 40 keeps the sound quality imperceptible, whereas the UQI improve this down to 35 clusters. The Max method fails to provide any interesting results.



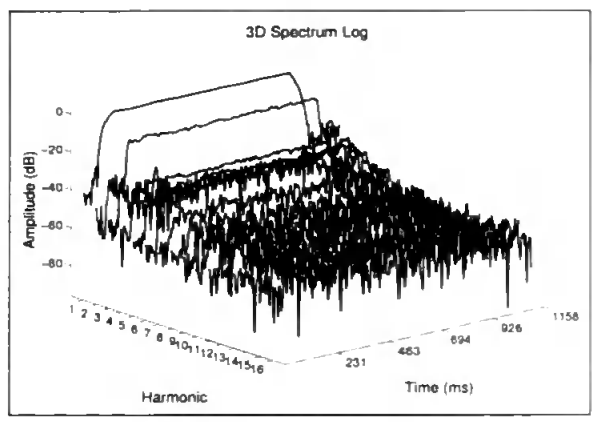
(a) Waveform



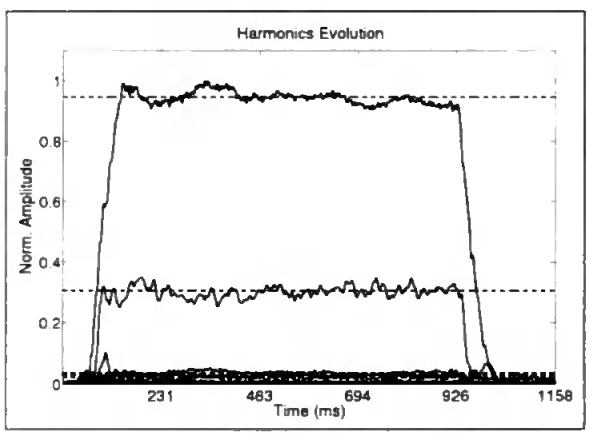
(b) Spectrum



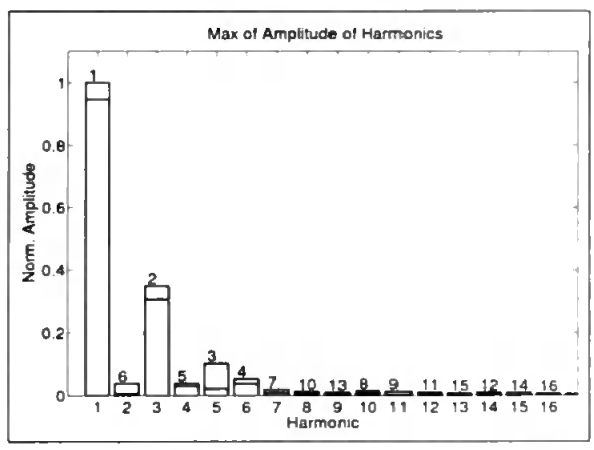
(c) Spectrum lines (Lin)



(d) Spectrum lines (Log)

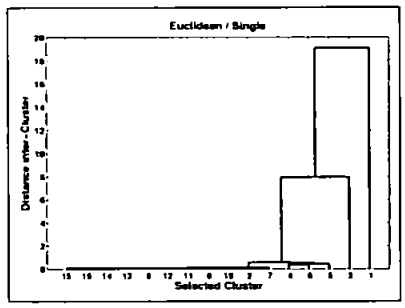


(e) Harmonics envelopes

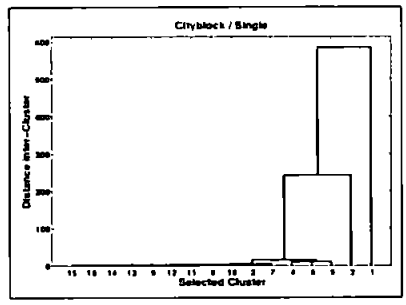


(f) Harmonics distribution

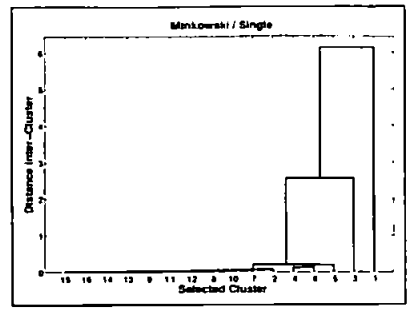
Figure 6.44: Principal - Sound analysis



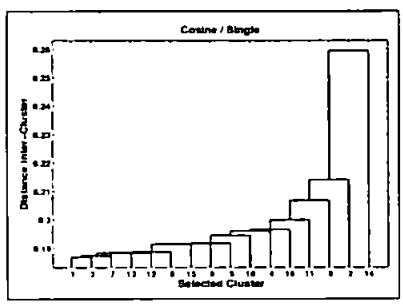
(a) Euclidean / Single



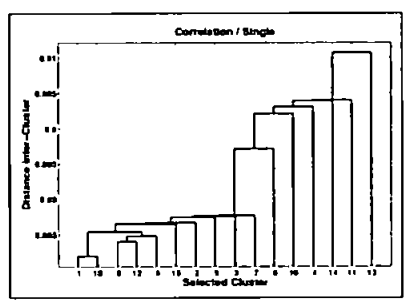
(b) Cityblock / Single



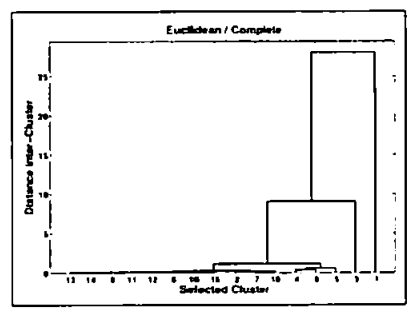
(c) Minkowski / Single



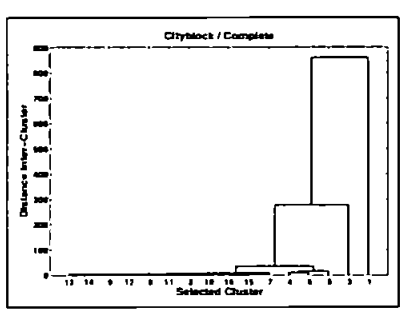
(d) Cosine / Single



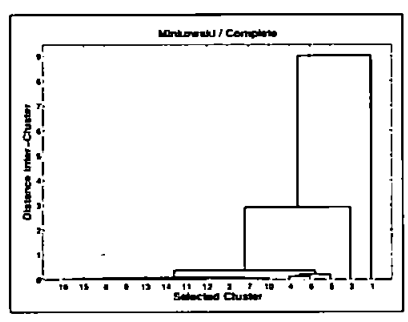
(e) Correlation / Single



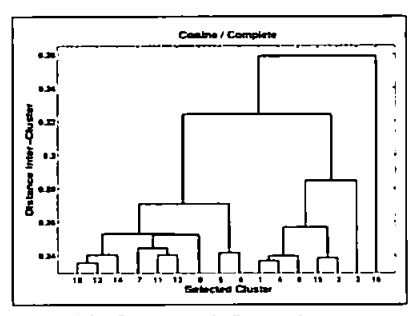
(f) Euclidean / Complete



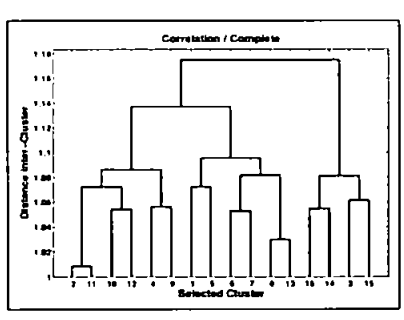
(g) Cityblock / Complete



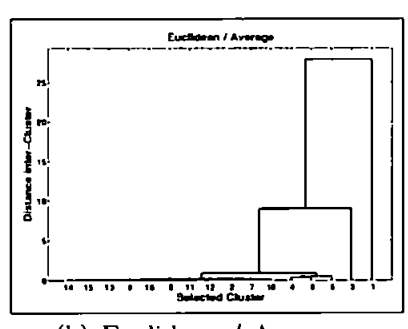
(h) Minkowski / Complete



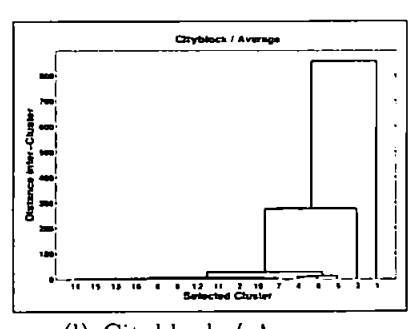
(i) Cosine / Complete



(j) Correlation / Complete

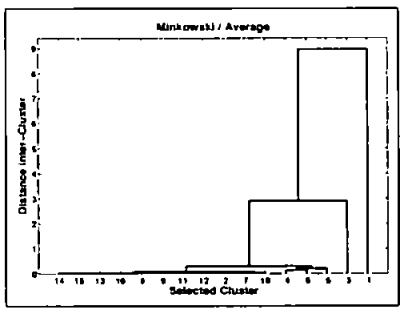


(k) Euclidean / Average

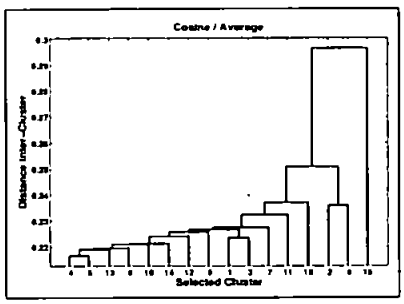


(l) Cityblock / Average

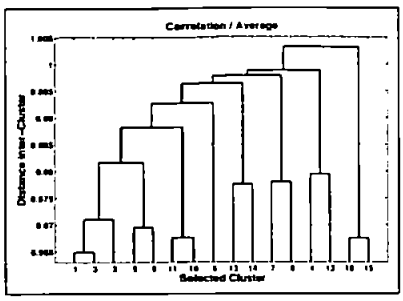
Figure 6.45: Principal - Dendrograms results



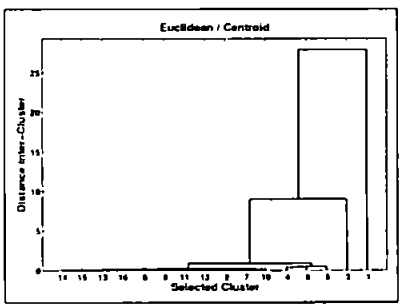
(a) Minkowski / Average



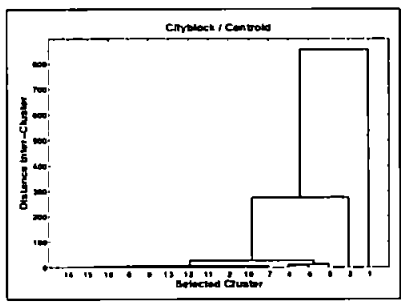
(b) Cosine / Average



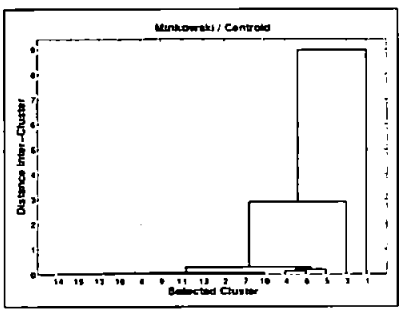
(c) Correlation / Average



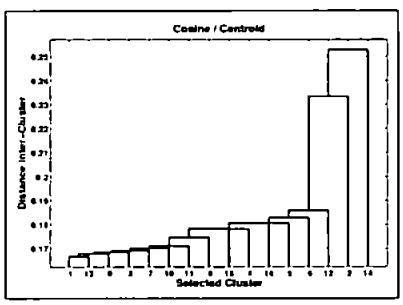
(d) Euclidean / Centroid



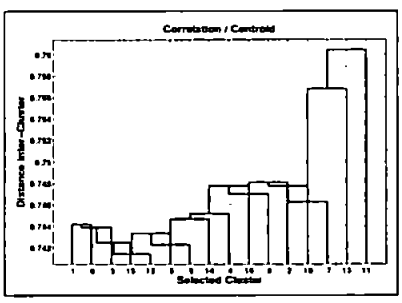
(e) Cityblock / Centroid



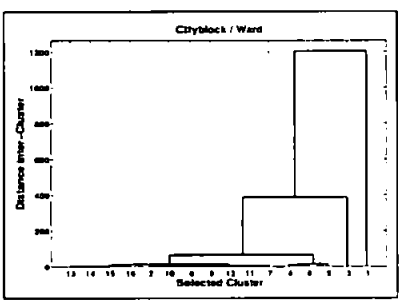
(f) Minkowski / Centroid



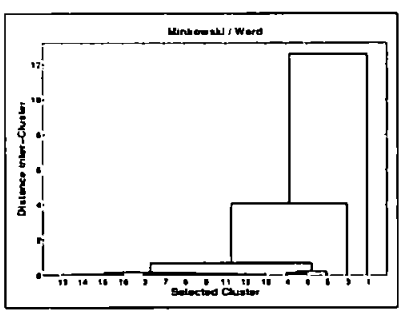
(g) Cosine / Centroid



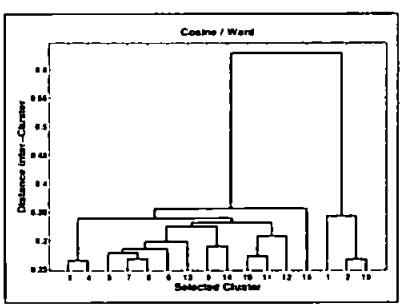
(h) Correlation / Centroid



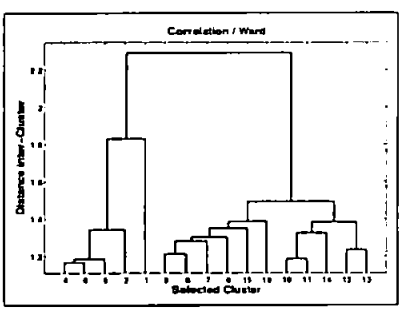
(i) Cityblock / Ward



(j) Minkowski / Ward

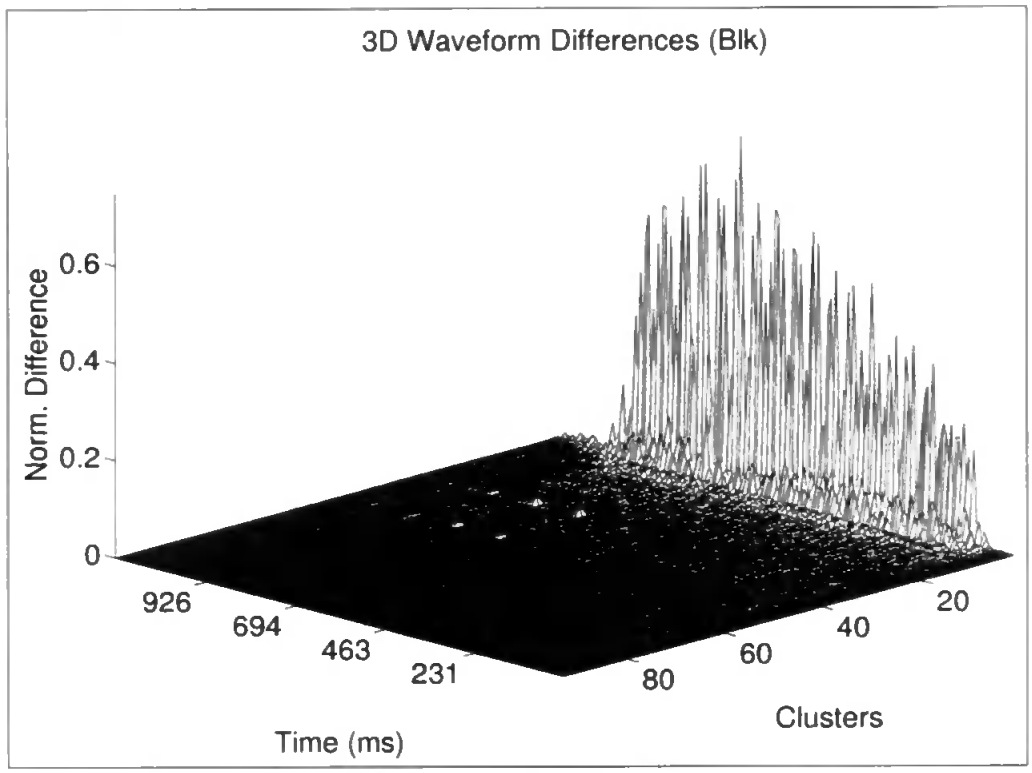


(k) Cosine / Ward

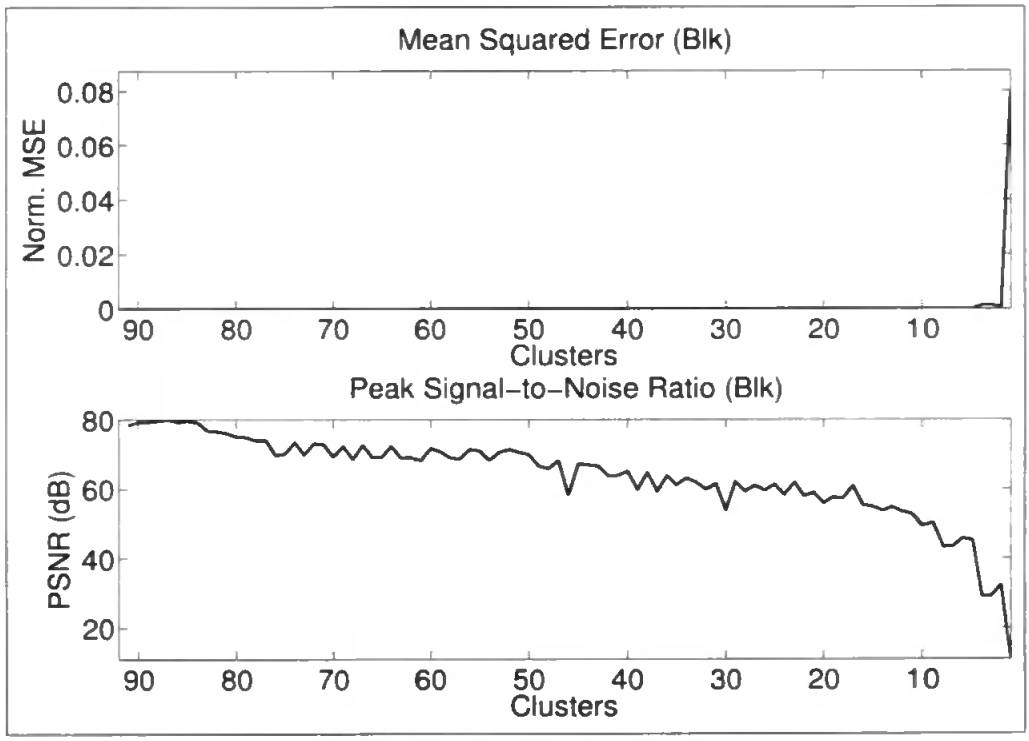


(l) Correlation / Ward

Figure 6.46: Principal - Dendrograms results (*continued*)

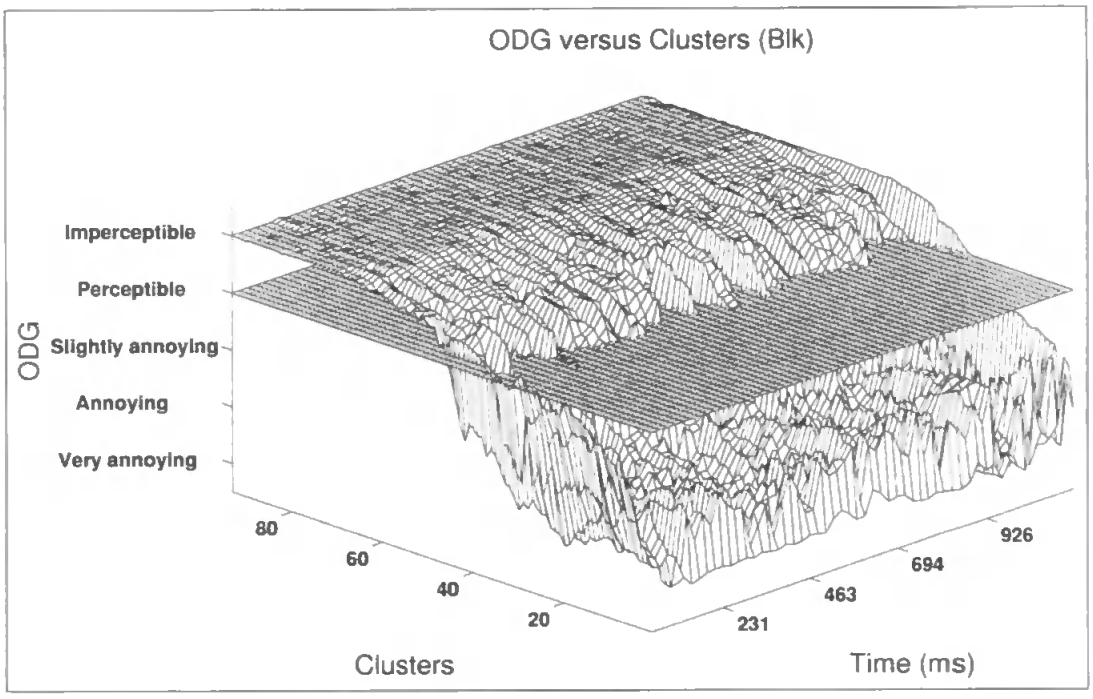


(a) Waveform modeling difference

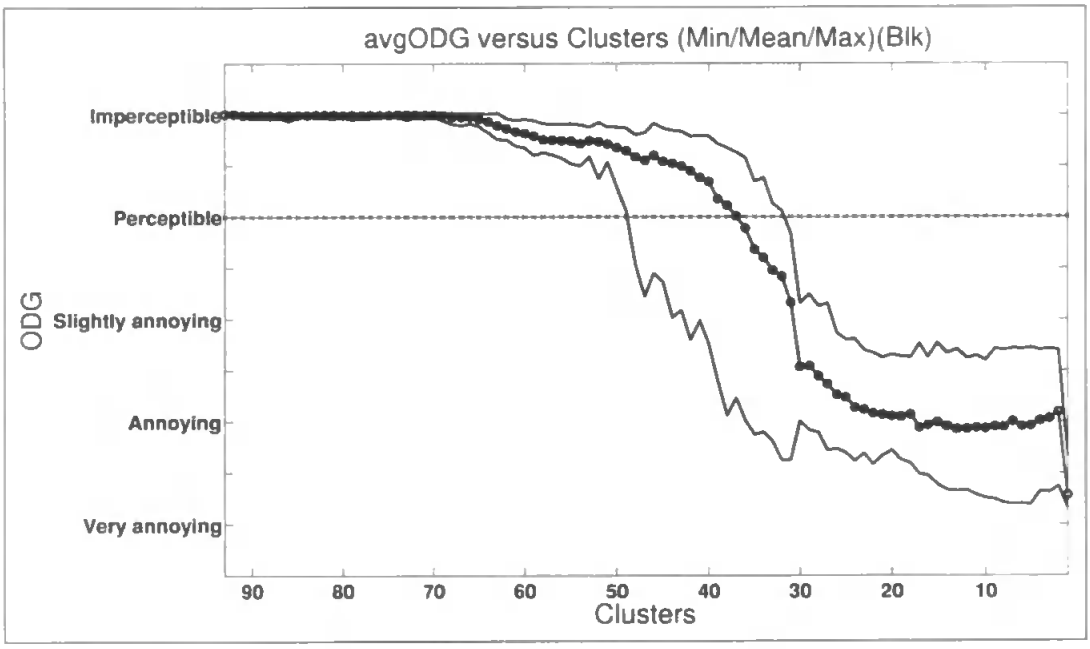


(b) MSE/PSNR Error modeling

Figure 6.47: Principal - Clustering results (Blk)

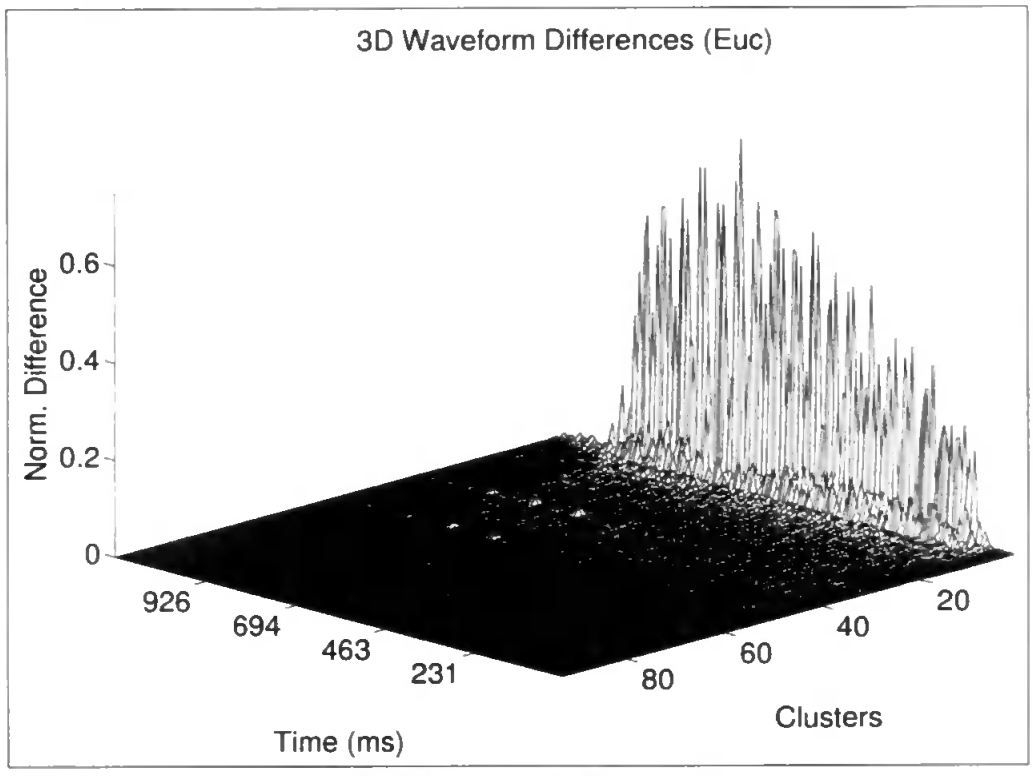


(a) ODGvsCL

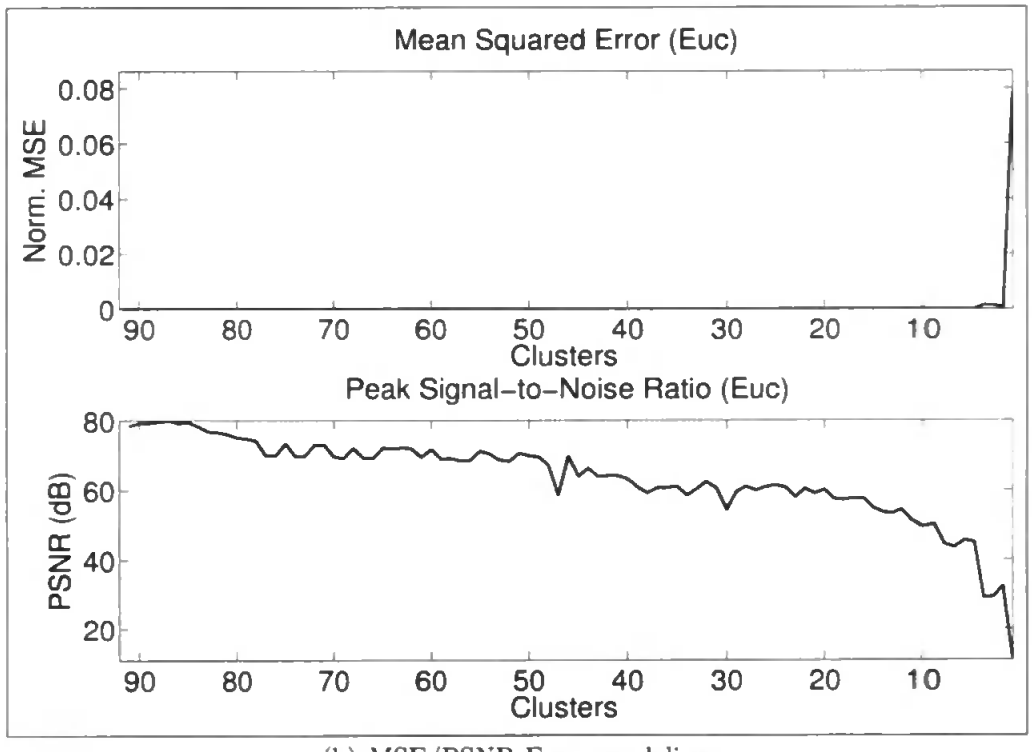


(b) avgODGvsCL

Figure 6.48: Principal - Clustering results ODG (Blk)

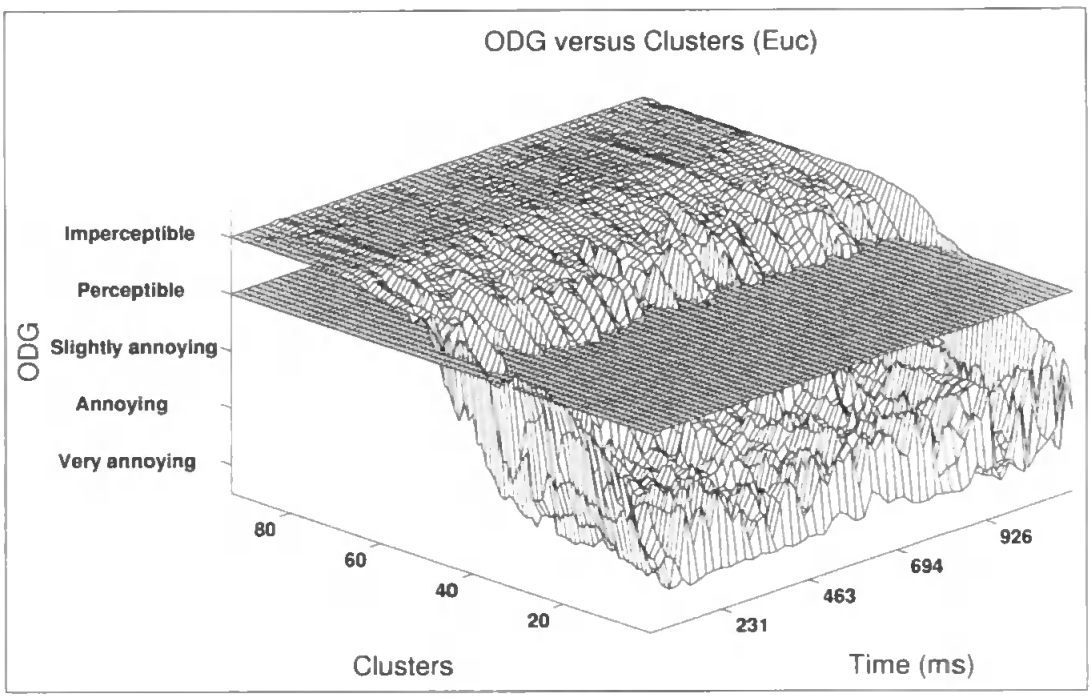


(a) Waveform modeling difference

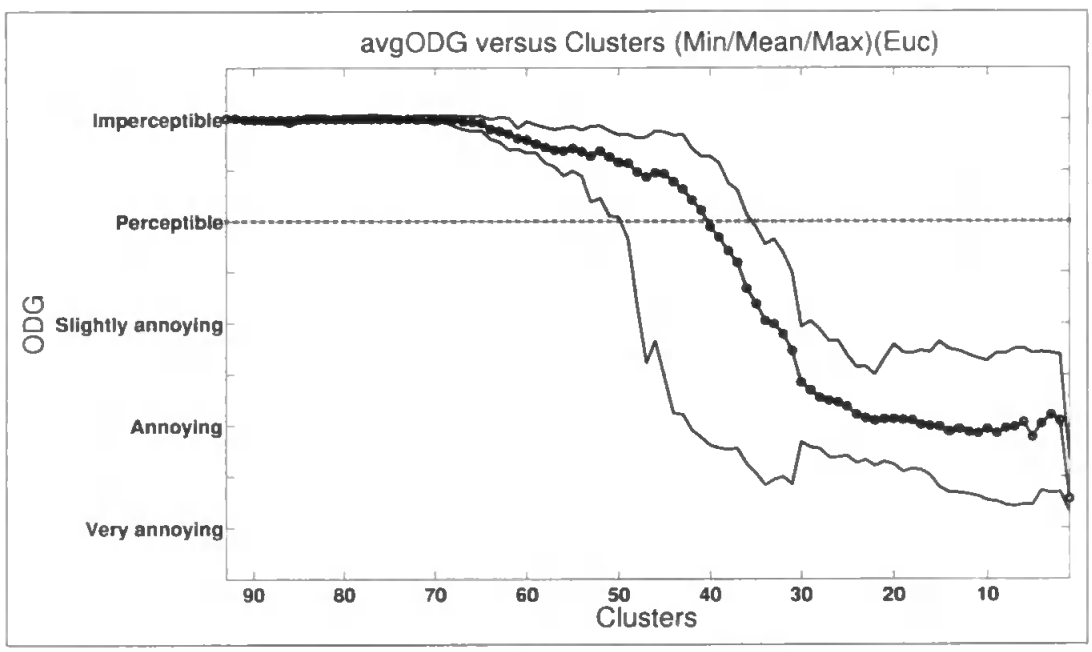


(b) MSE/PSNR Error modeling

Figure 6.49: Principal - Clustering results (Euc)

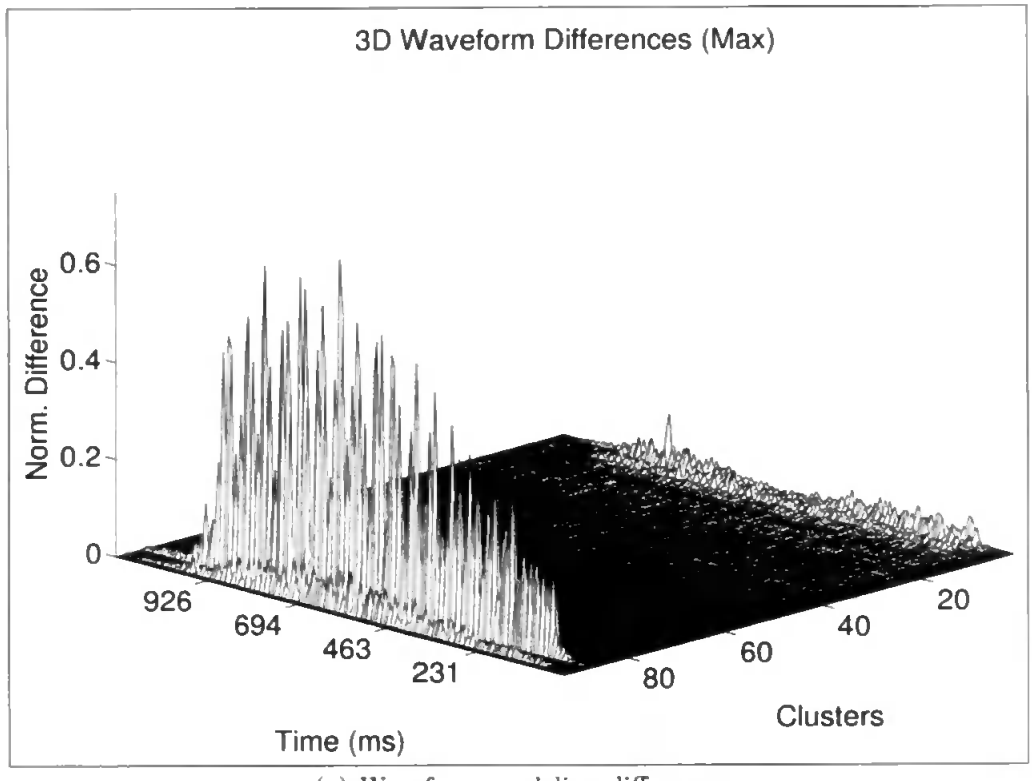


(a) ODGvsCL

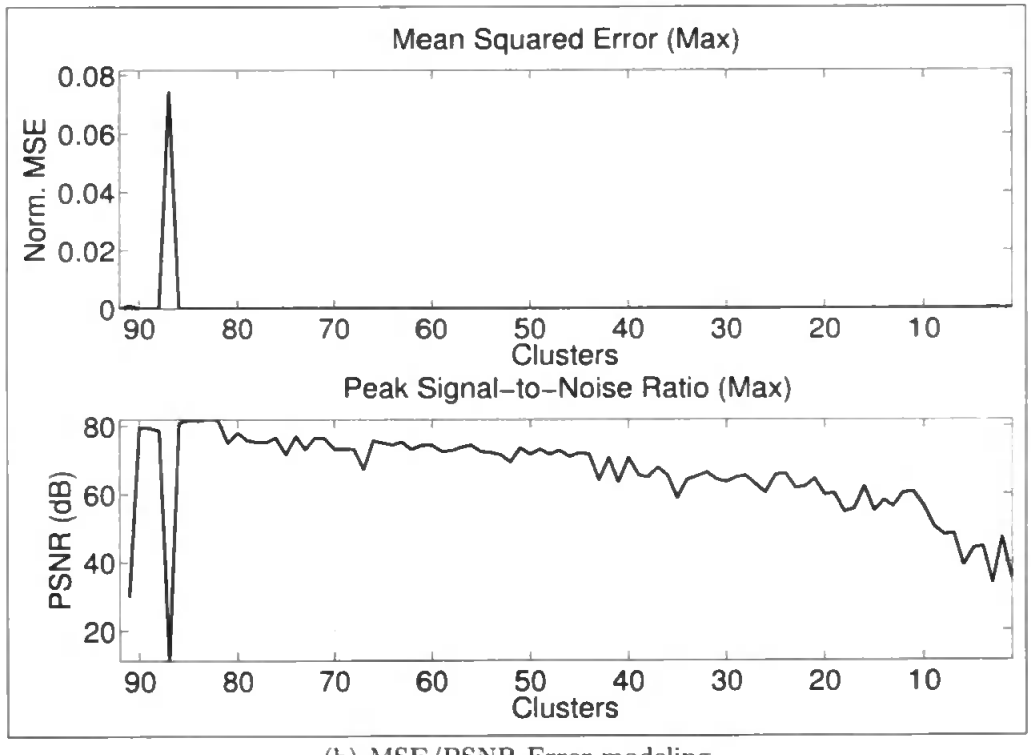


(b) avgODGvsCL

Figure 6.50: Principal - Clustering results ODG (Euc)

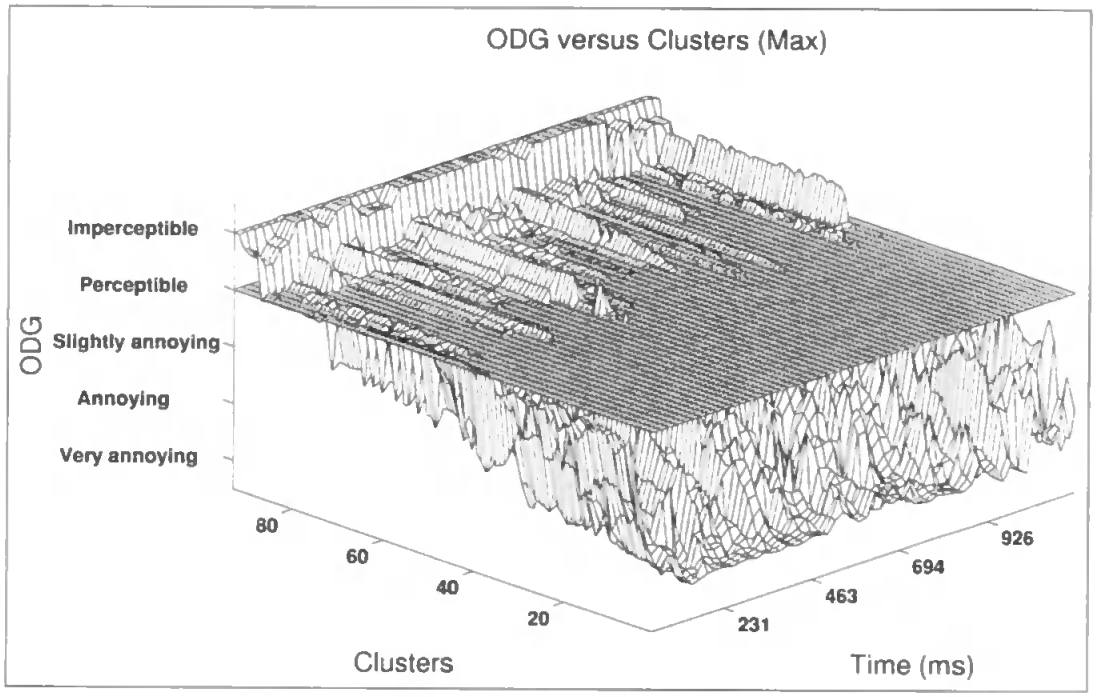


(a) Waveform modeling difference

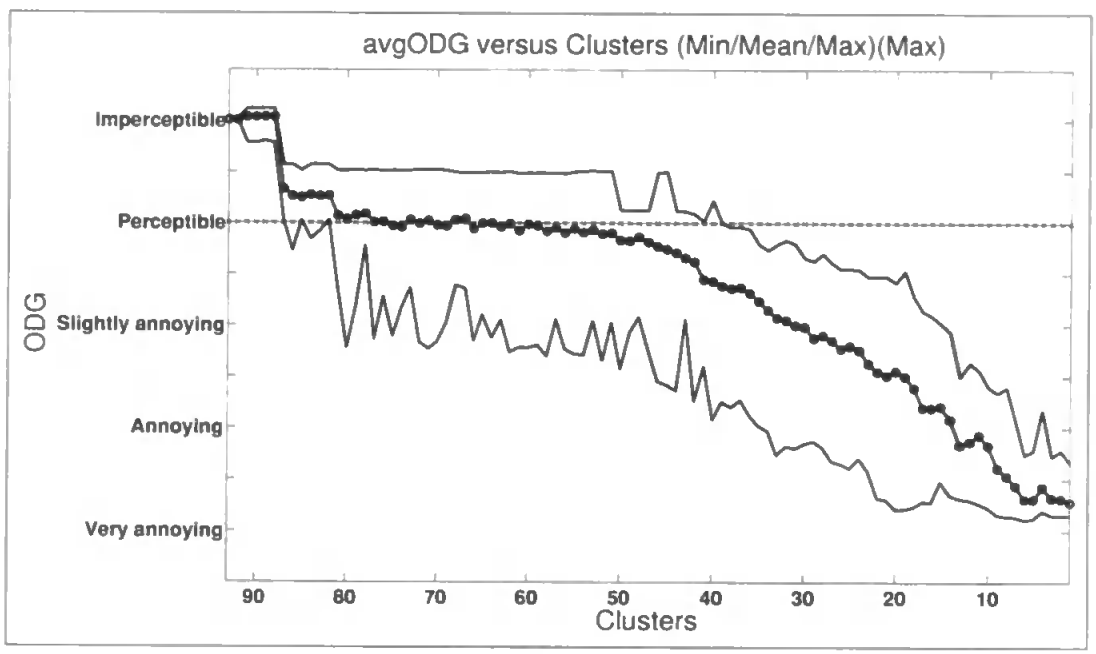


(b) MSE/PSNR Error modeling

Figure 6.51: Principal - Clustering results (Max)

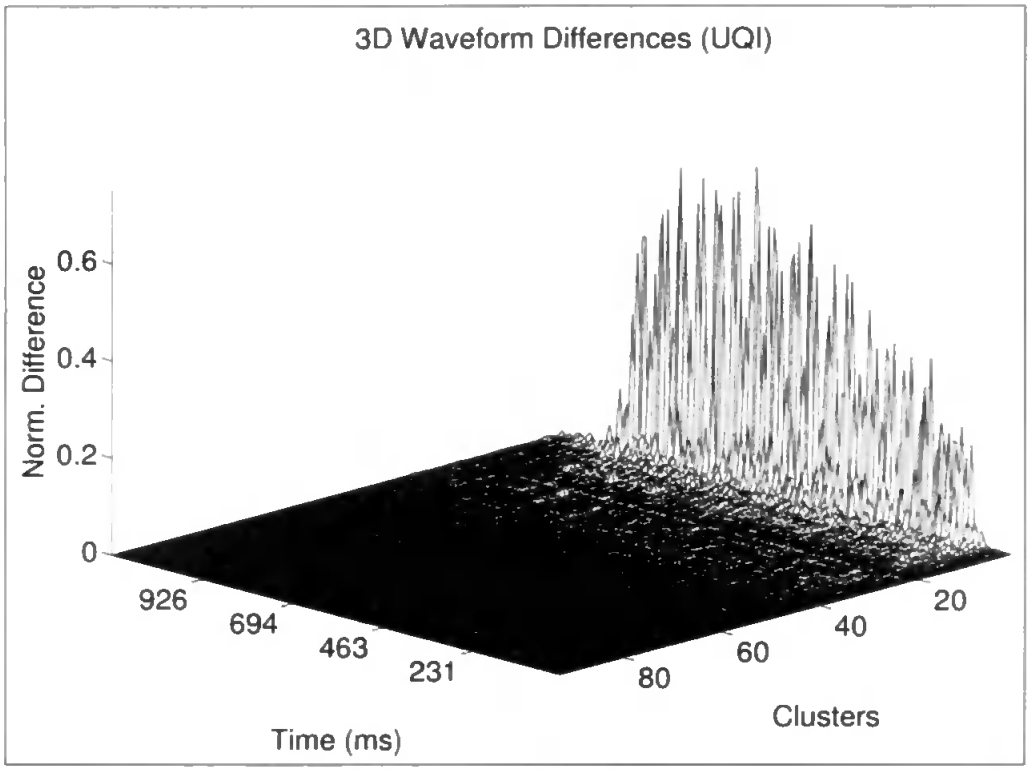


(a) ODGvsCL

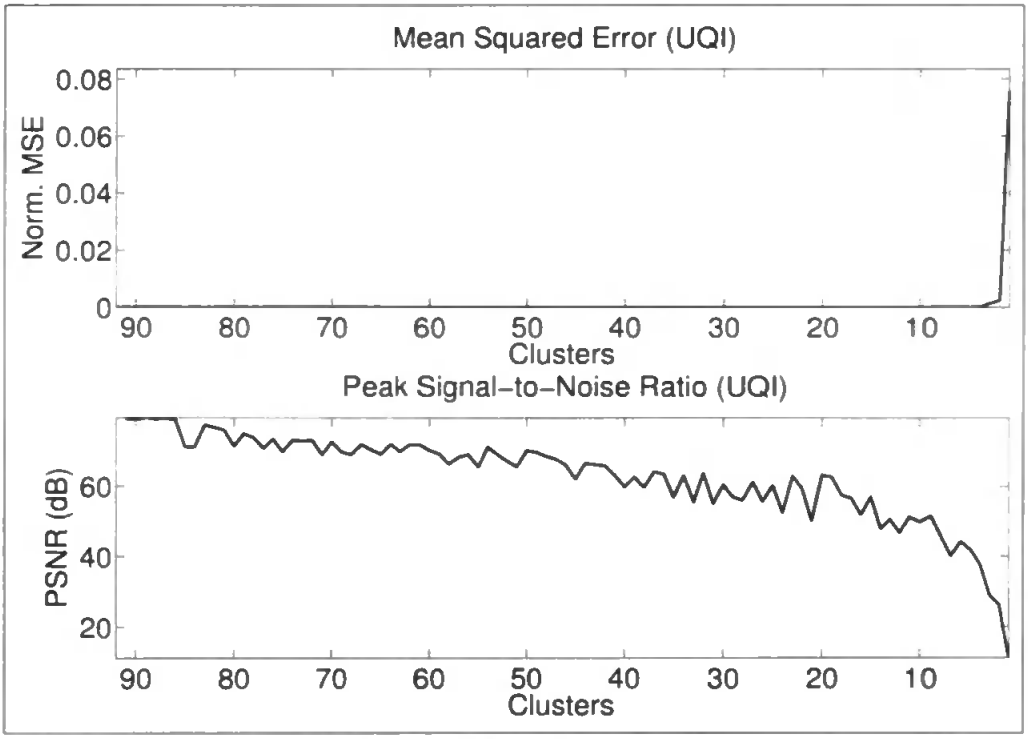


(b) avgODGvsCL

Figure 6.52: Principal - Clustering results ODG (Max)

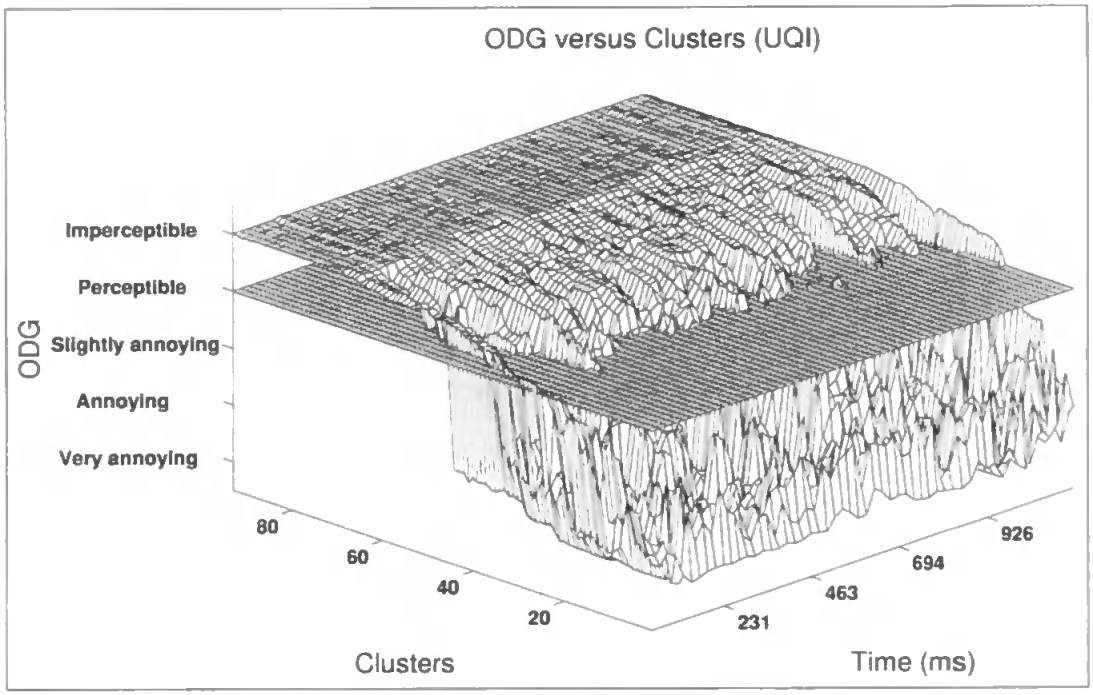


(a) Waveform modeling difference

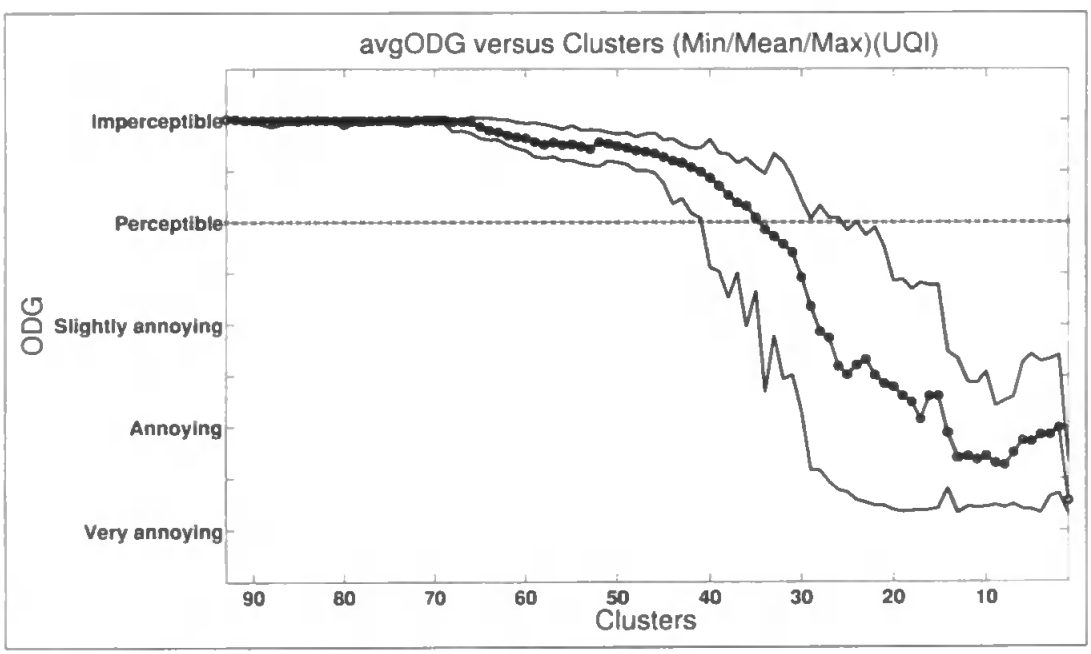


(b) MSE/PSNR Error modeling

Figure 6.53: Principal - Clustering results (UQI)



(a) ODGvsCL



(b) avgODGvsCL

Figure 6.54: Principal - Clustering results ODG (UQI)

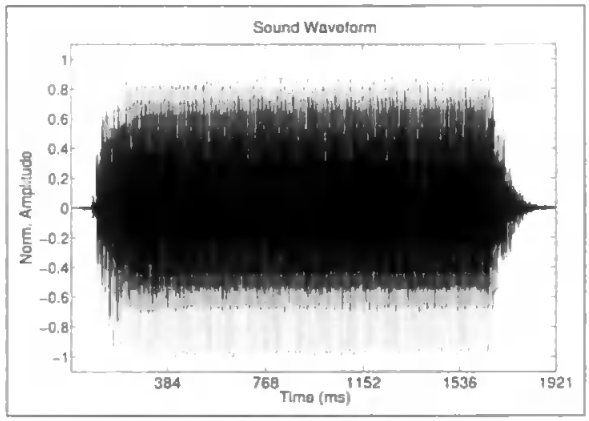
6.4.7 Sound - Metal Trombone 16'

Sound Analysis - This sound has been recorded from a Metal Trombone 16' pipe and has a fundamental of about 131 Hz. The sound's waveform is shown in Figure 6.55(a) and its spectrum in Figure 6.55(b). Figure 6.55(c) and Figure 6.55(d) shown the spectrum with lines, compared to previous sound examples, it is clear that this sound has a much more complex harmonics structure. This is confirmed with Figure 6.55(e) with all the evolution of the harmonics super-imposed, and Figure 6.55(f) showing its harmonics distribution. Many harmonics in Figure 6.55(e) have high sustain level. Figure 6.55(f) shown that the harmonic distribution has clearly a lot of high harmonics with the highest being the 6^{th}, the 5^{th}, the 8^{th}, the 11^{th}, the 4^{th}, and the 12^{th} in this order (indicated a the top of the bar graph). Notice also the variation in amplitude of many of these harmonics.

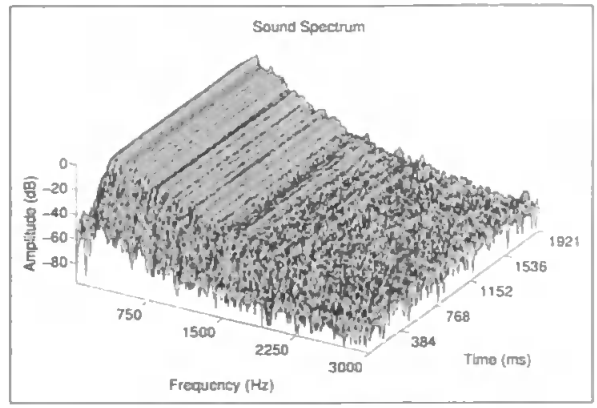
Dendrogram - The Agglomerative Hierarchical Clustering (AHC) algorithm, as described in Section 3.3.5, with different methods for calculating the metric distance and linkage methods, results in Dendrograms as shown in Figures 6.56(a) to 6.56(l) and Figures 6.57(a) to 6.57(l).

Modeling Error - The modeling errors results of the AHC algorithm using 4 distance metric methods (Cityblock, Euclidean, Max and UQI) are shown in Figure 6.47, Figure 6.49, Figure 6.51 and Figure 6.53

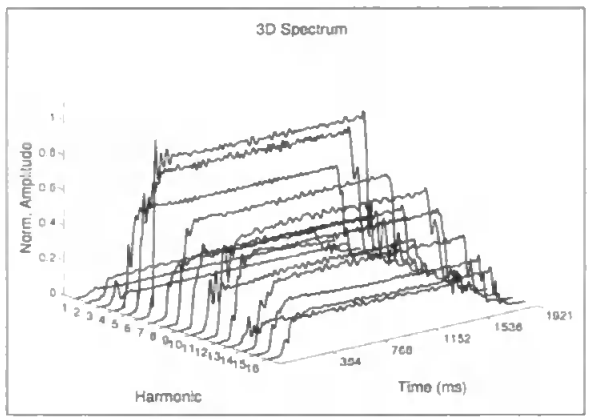
Objective Audio Quality - Figure 6.47, Figure 6.49, Figure 6.51 and Figure 6.53 are showing the objective audio quality of the AHC algorithm using 4 distance metric methods (Cityblock, Euclidean, Max and UQI). Clearly the algorithm struggle to cluster properly the harmonics as all four cases cut the perceptible line at a very high number of clusters. This sound is a real modeling challenge from its harmonic structure.



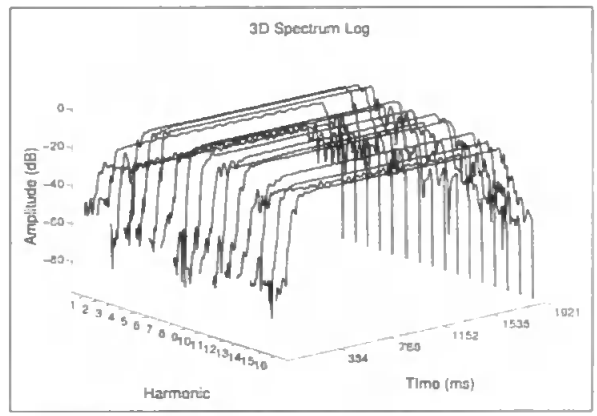
(a) Waveform



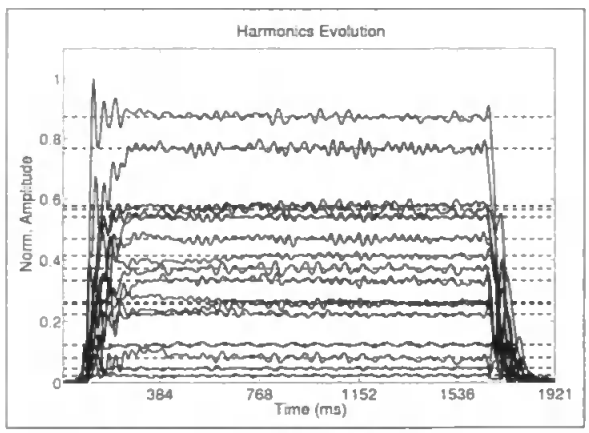
(b) Spectrum



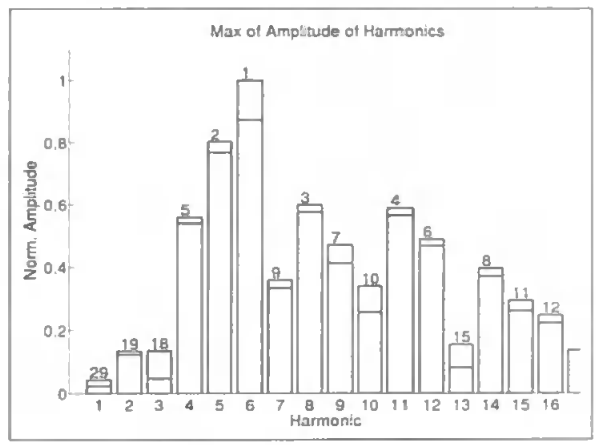
(c) Spectrum lines (Lin)



(d) Spectrum lines (Log)

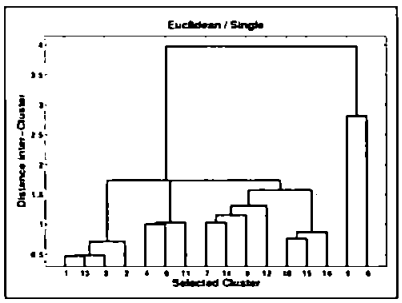


(e) Harmonics envelopes

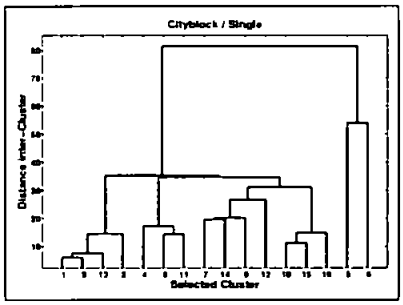


(f) Harmonics distribution

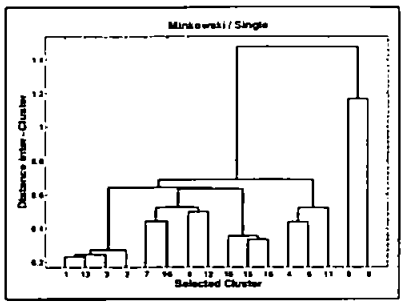
Figure 6.55: Trombone - Sound analysis



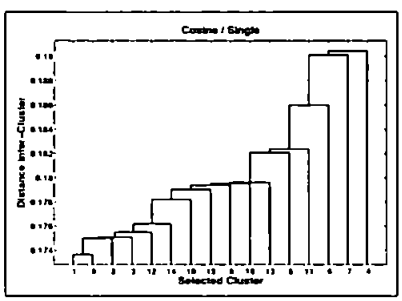
(a) Euclidean / Single



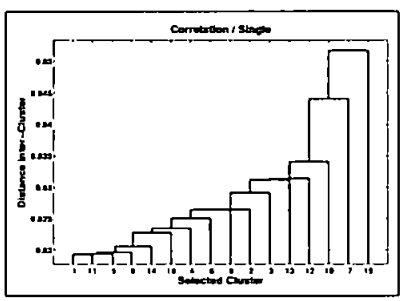
(b) Cityblock / Single



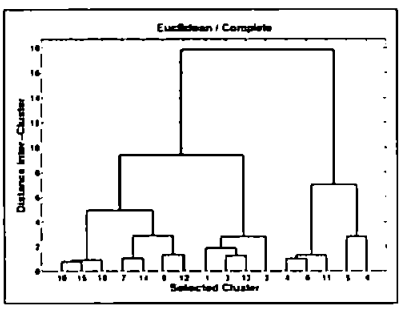
(c) Minkowski / Single



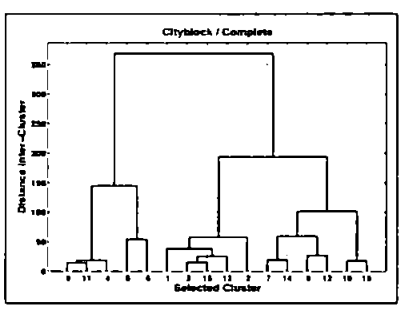
(d) Cosine / Single



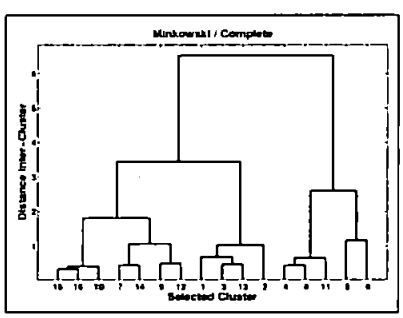
(e) Correlation / Single



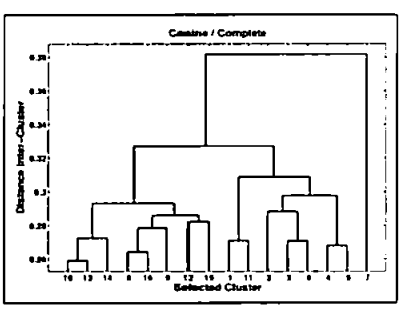
(f) Euclidean / Complete



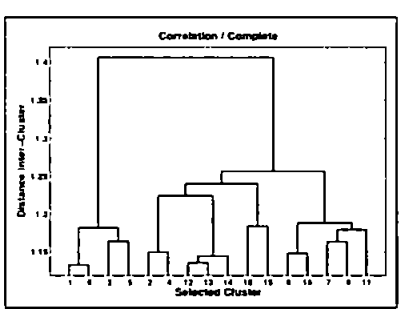
(g) Cityblock / Complete



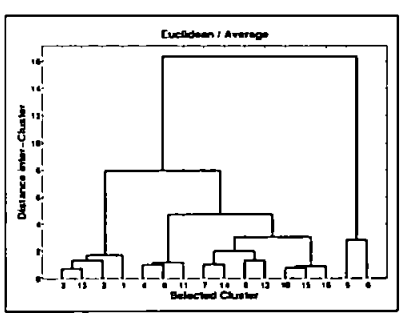
(h) Minkowski / Complete



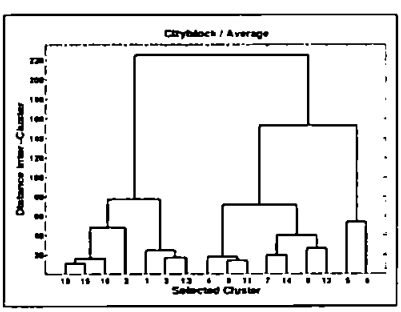
(i) Cosine / Complete



(j) Correlation / Complete

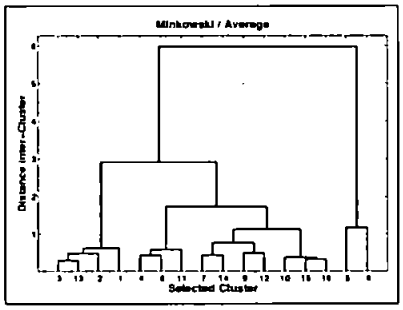


(k) Euclidean / Average

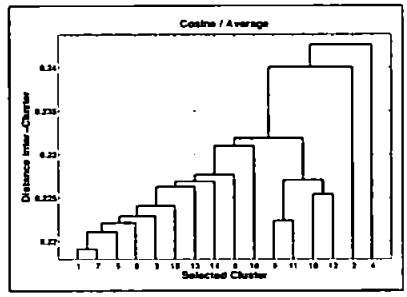


(l) Cityblock / Average

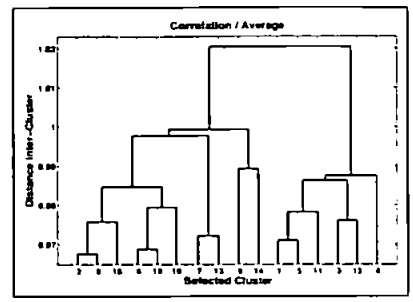
Figure 6.56: Trombone - Dendrogram results



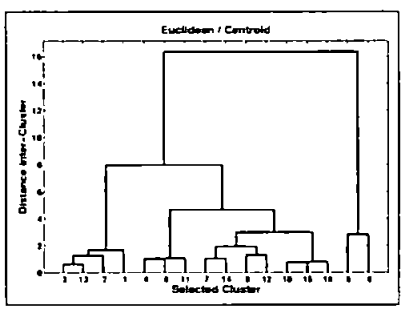
(a) Minkowski / Average



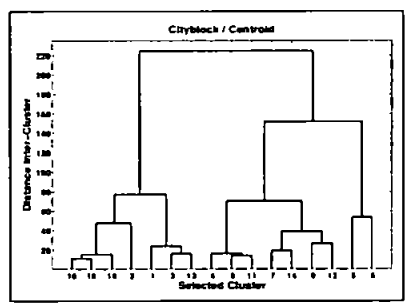
(b) Cosine / Average



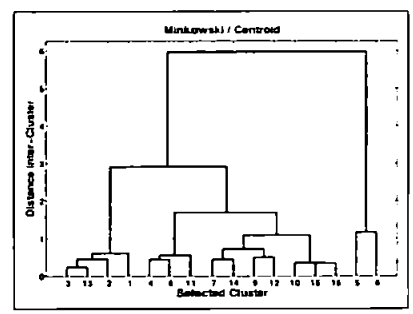
(c) Correlation / Average



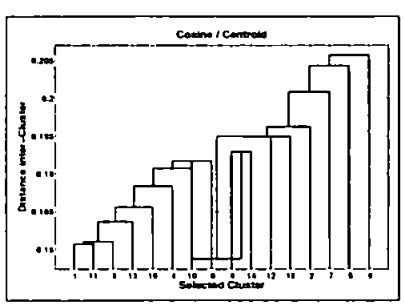
(d) Euclidean / Centroid



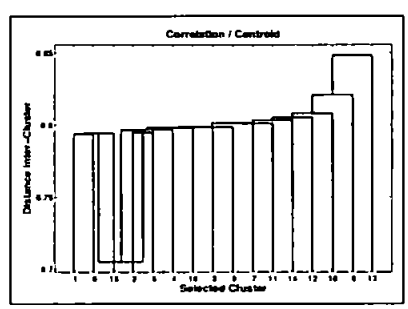
(e) Cityblock / Centroid



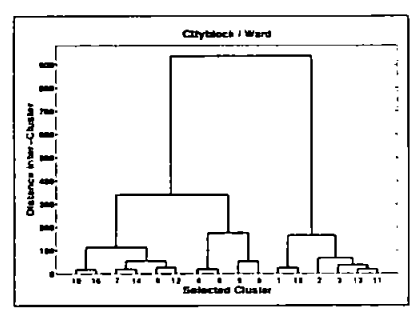
(f) Minkowski / Centroid



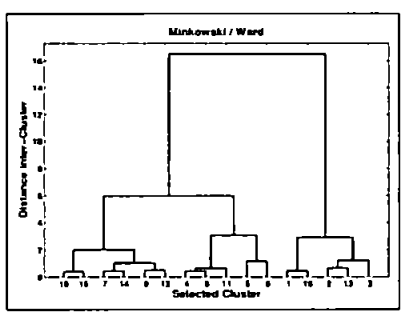
(g) Cosine / Centroid



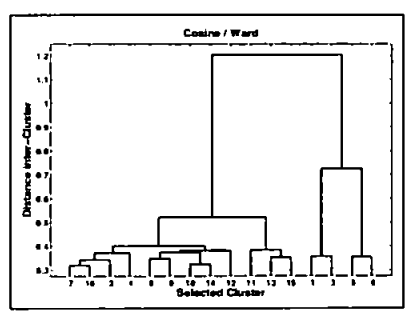
(h) Correlation / Centroid



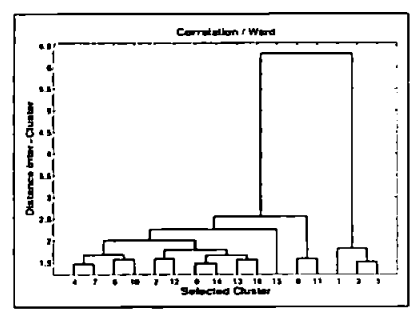
(i) Cityblock / Ward



(j) Minkowski / Ward

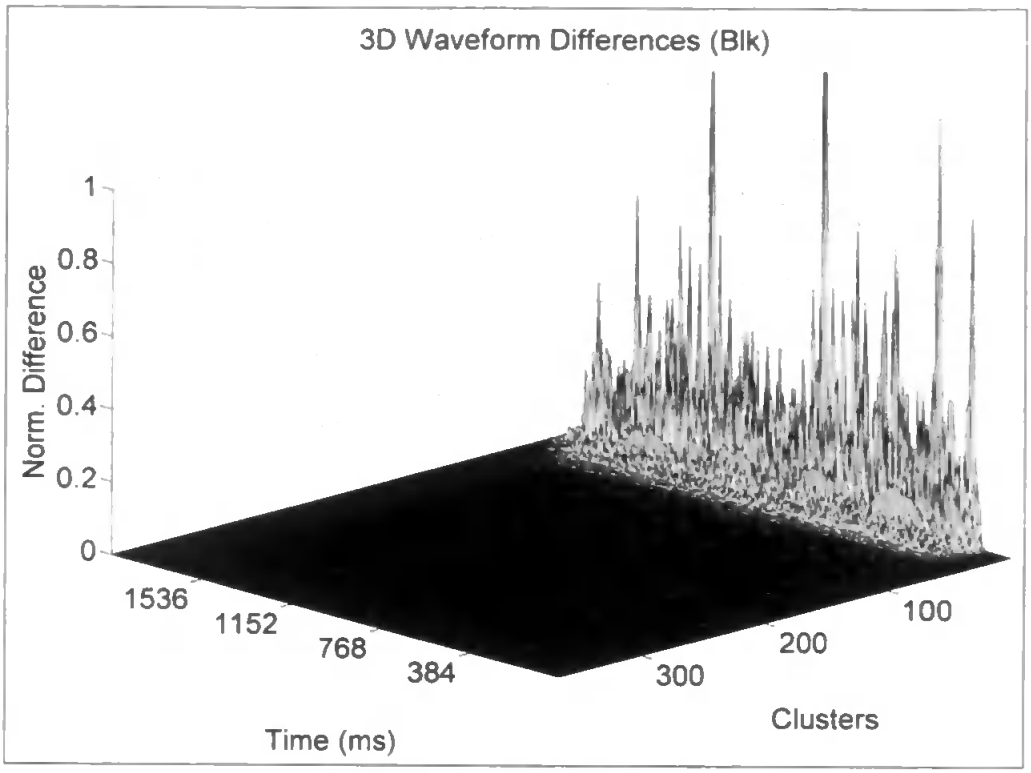


(k) Cosine / Ward

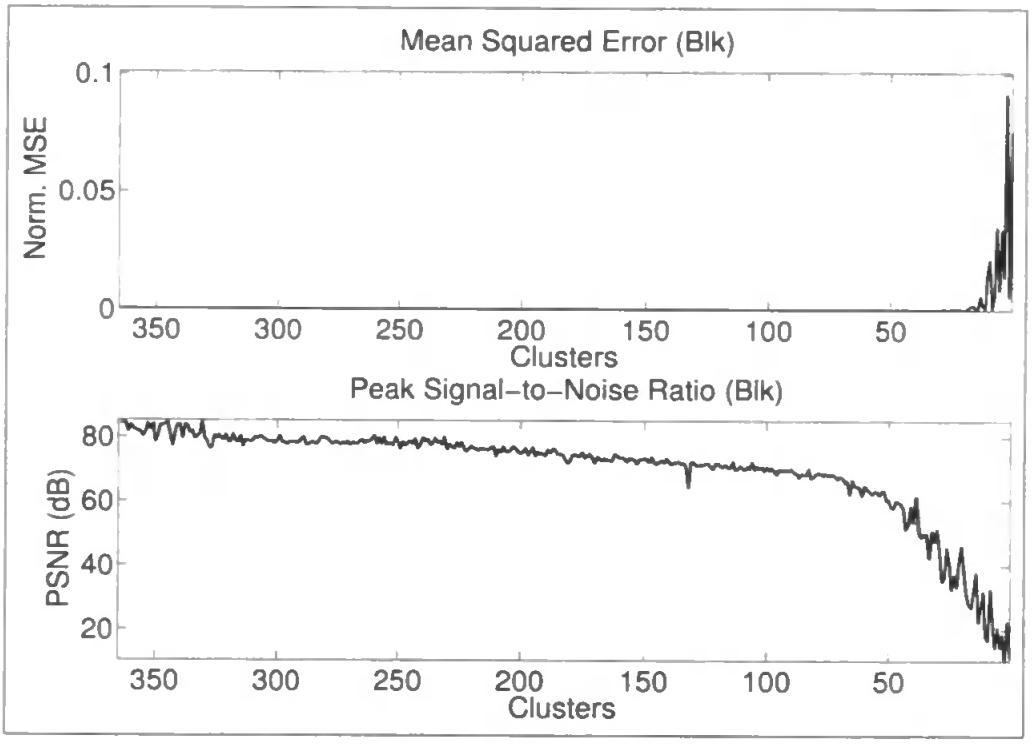


(l) Correlation / Ward

Figure 6.57: Trombone - Dendrogram results (*continued*)

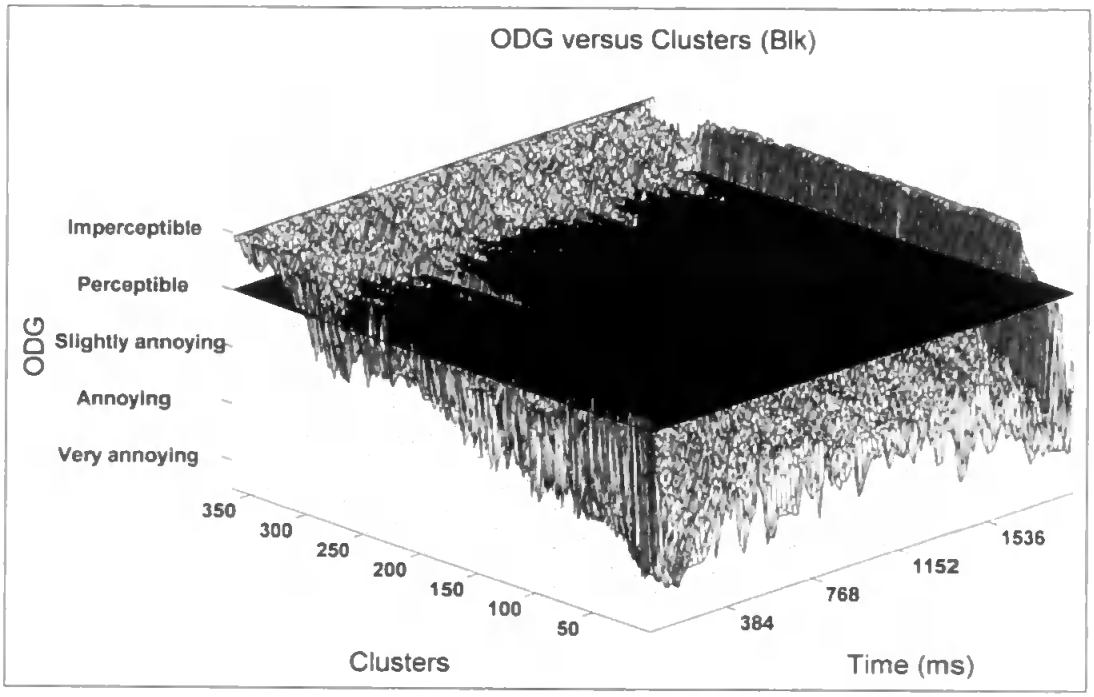


(a) Waveform modeling difference

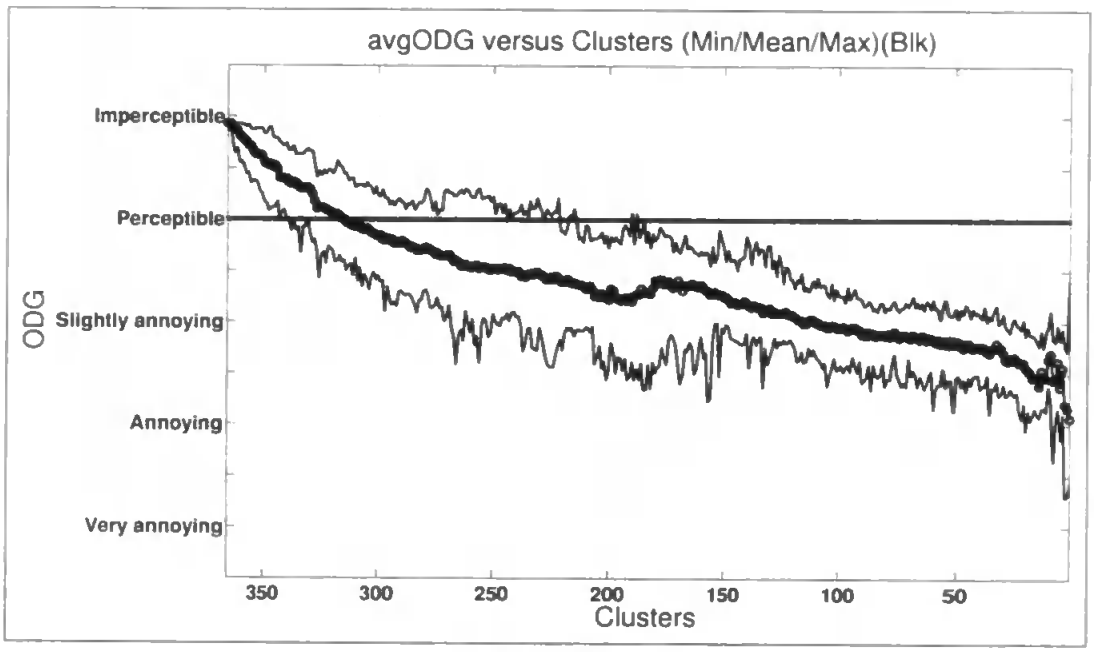


(b) MSE/PSNR Error modeling

Figure 6.58: Trombone - Clustering results (Blk)

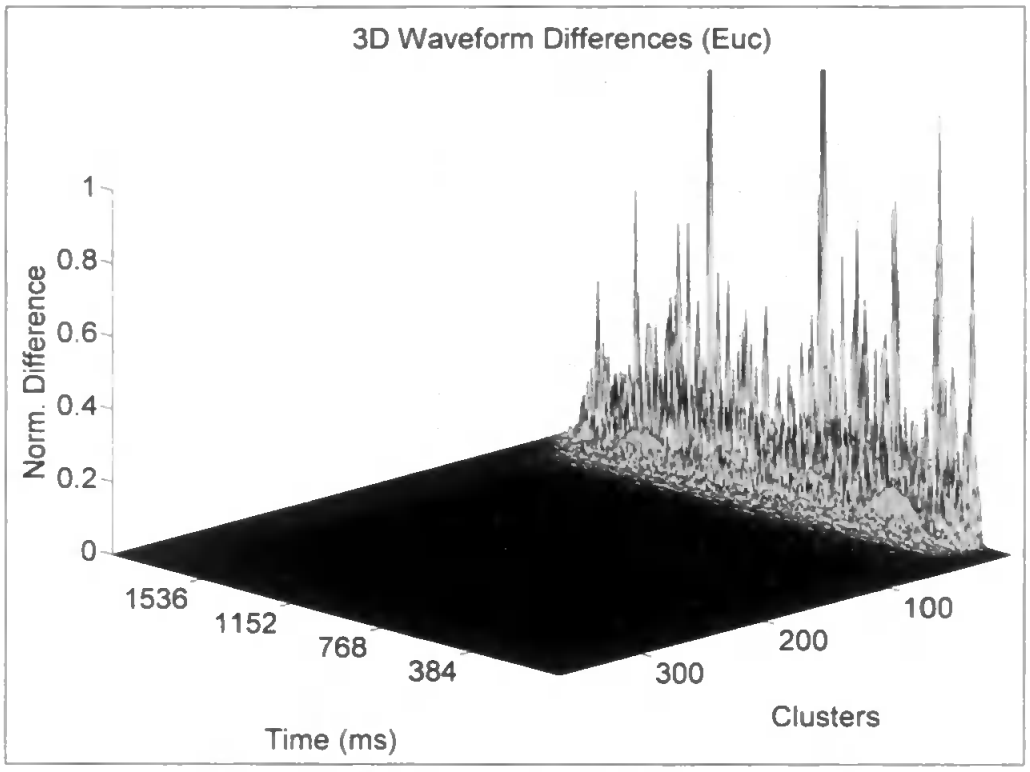


(a) ODGvsCL

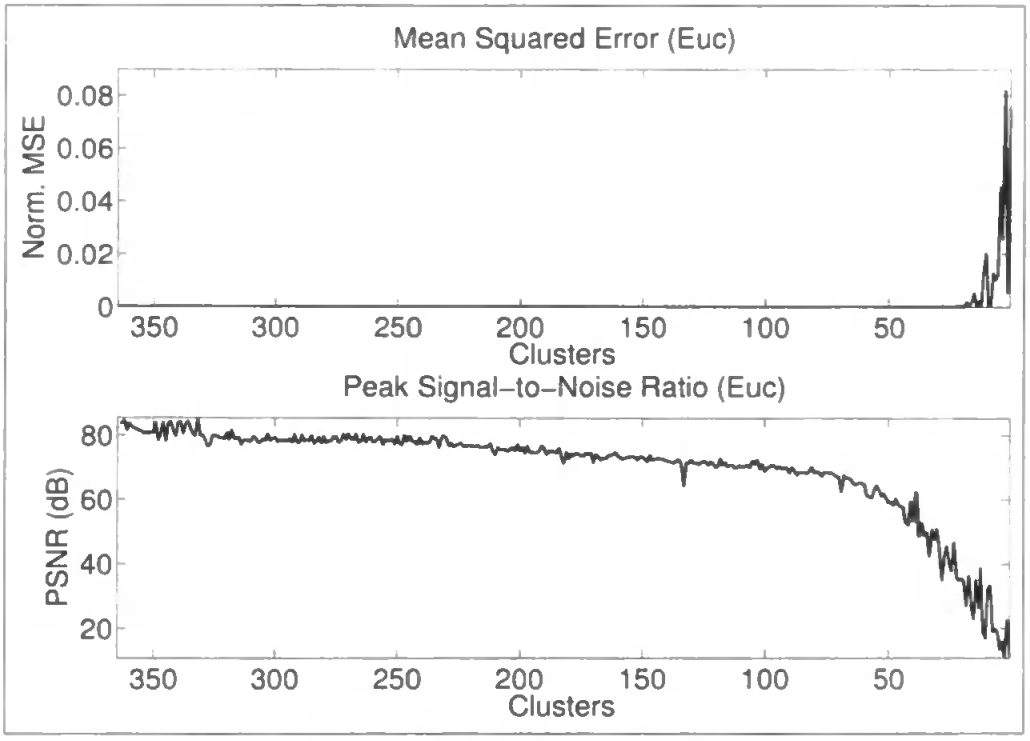


(b) avgODGvsCL

Figure 6.59: Trombone - Clustering results ODG (Blk)

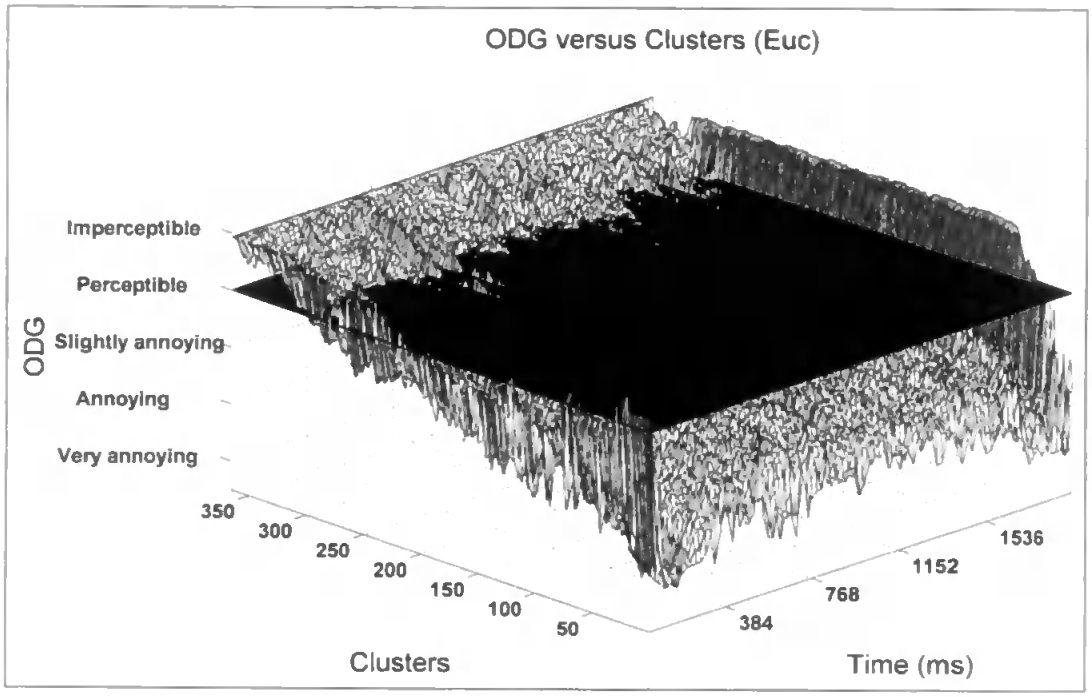


(a) Waveform modeling difference

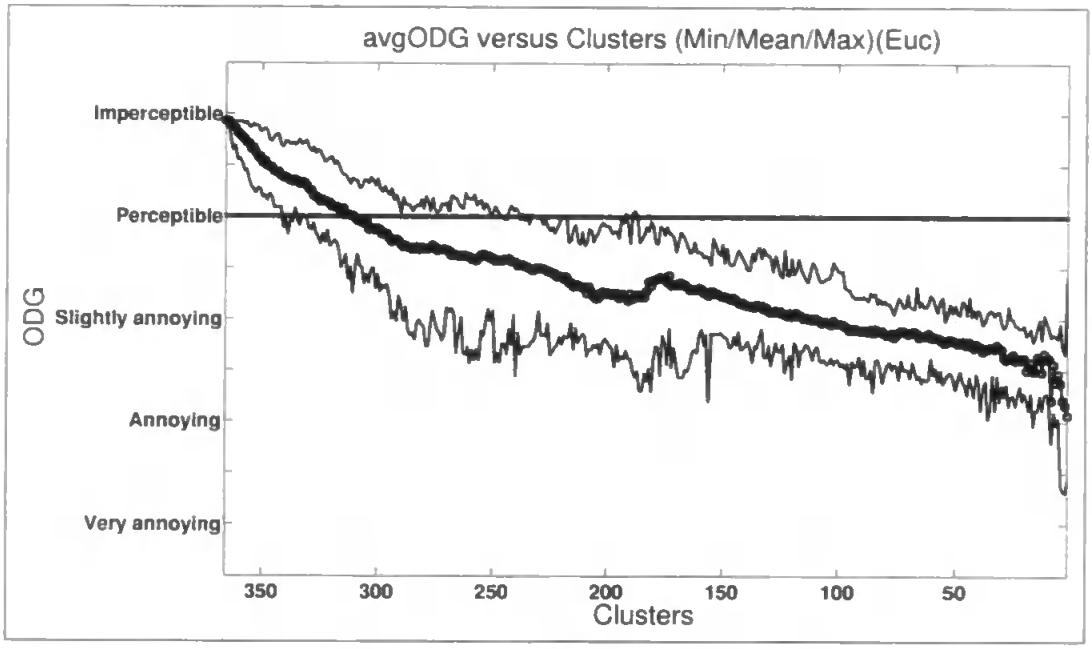


(b) MSE/PSNR Error modeling

Figure 6.60: Trombone - Clustering results (Euc)

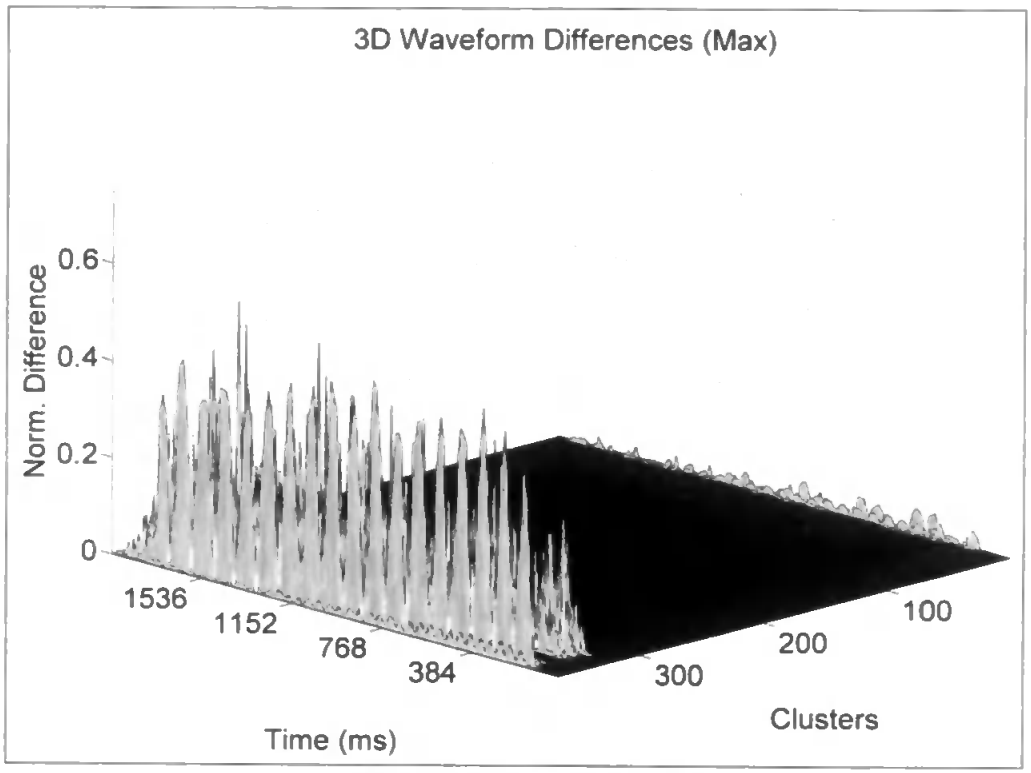


(a) ODGvsCL

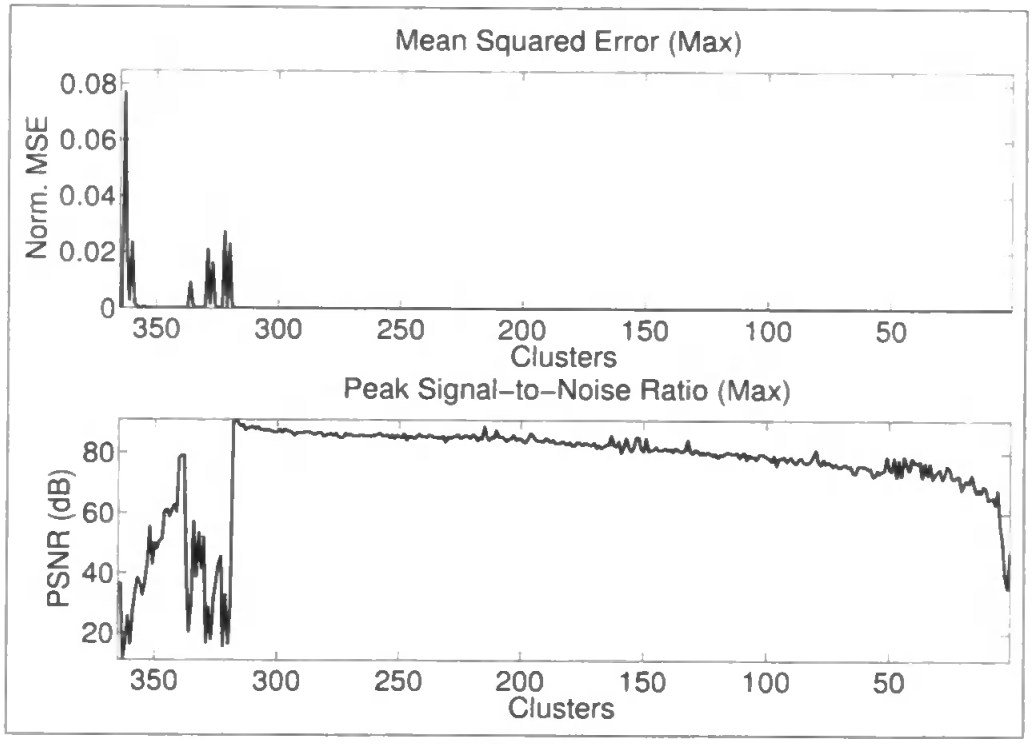


(b) avgODGvsCL

Figure 6.61: Trombone - Clustering results ODG (Euc)

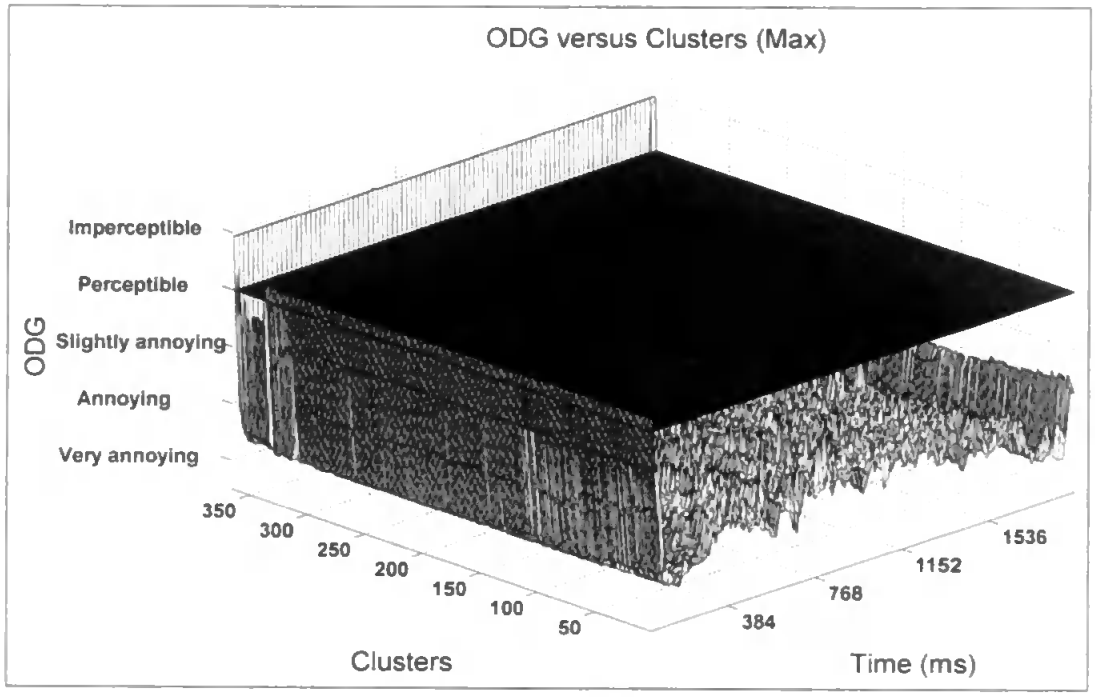


(a) Waveform modeling difference

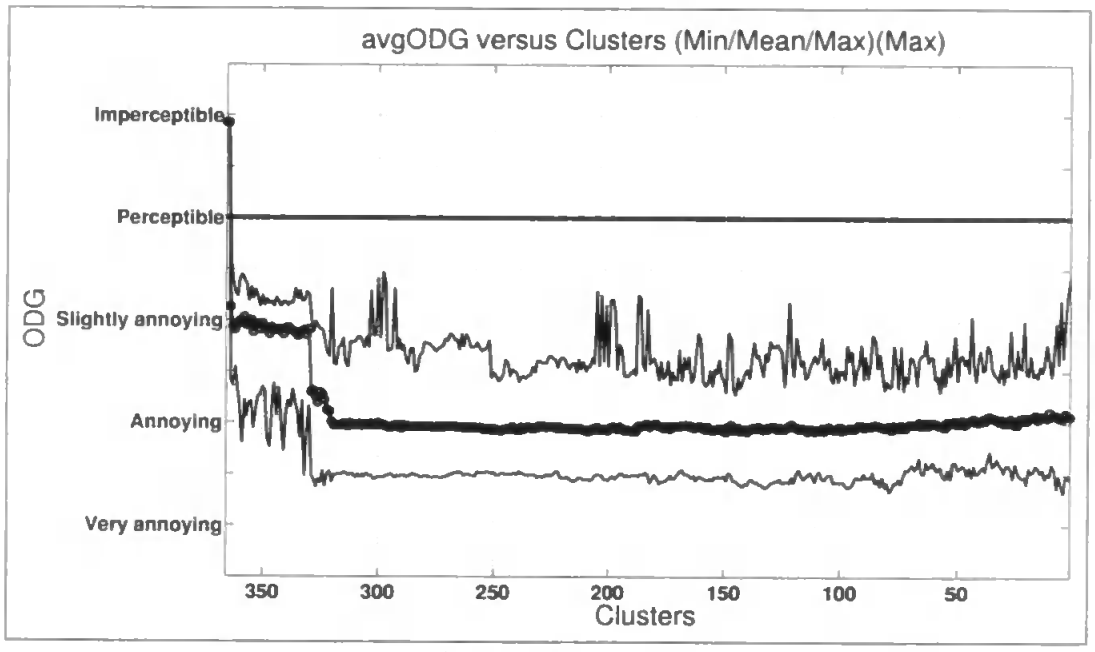


(b) MSE/PSNR Error modeling

Figure 6.62: Trombone - Clustering results (Max)

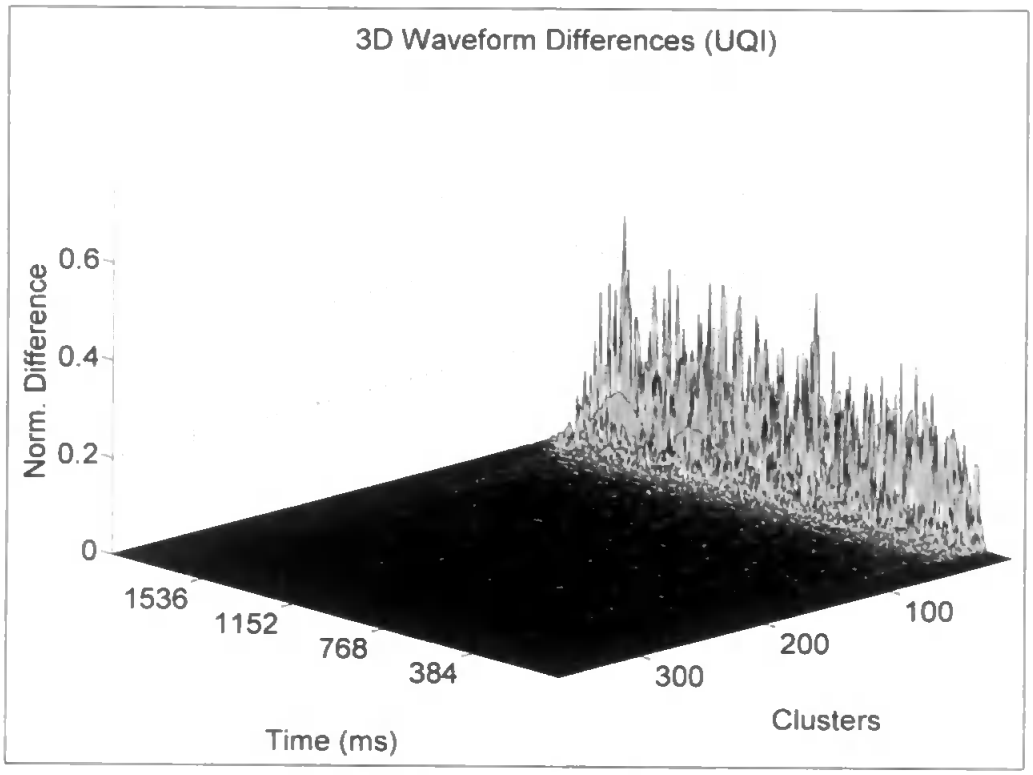


(a) ODGvsCL

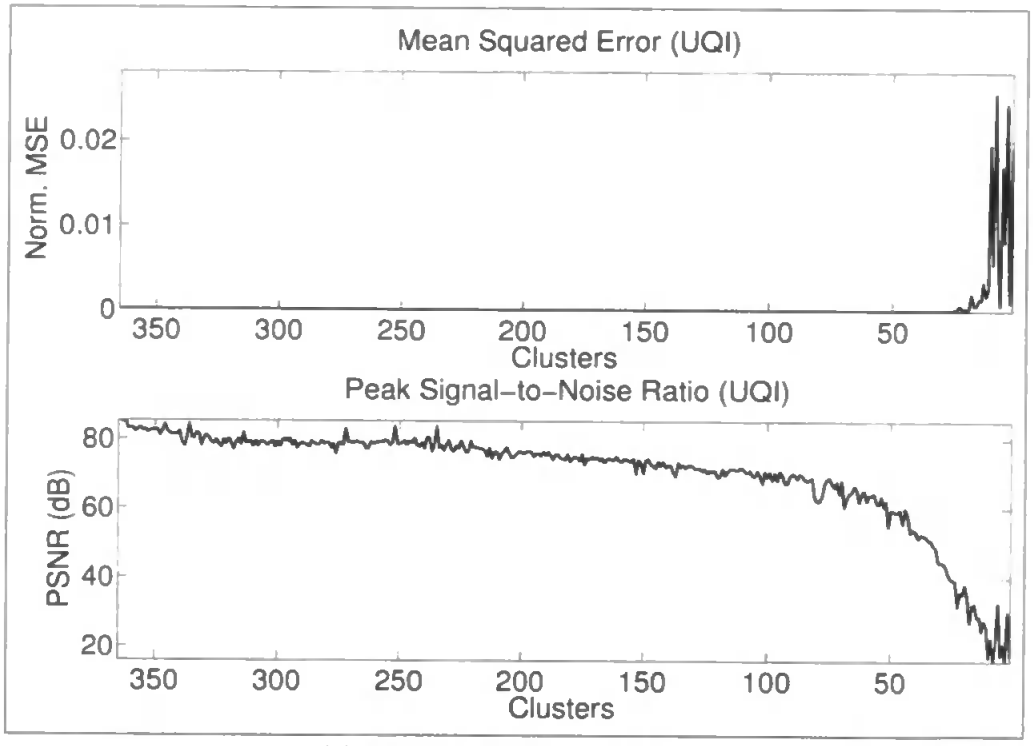


(b) avgODGvsCL

Figure 6.63: Trombone - Clustering results ODG (Max)

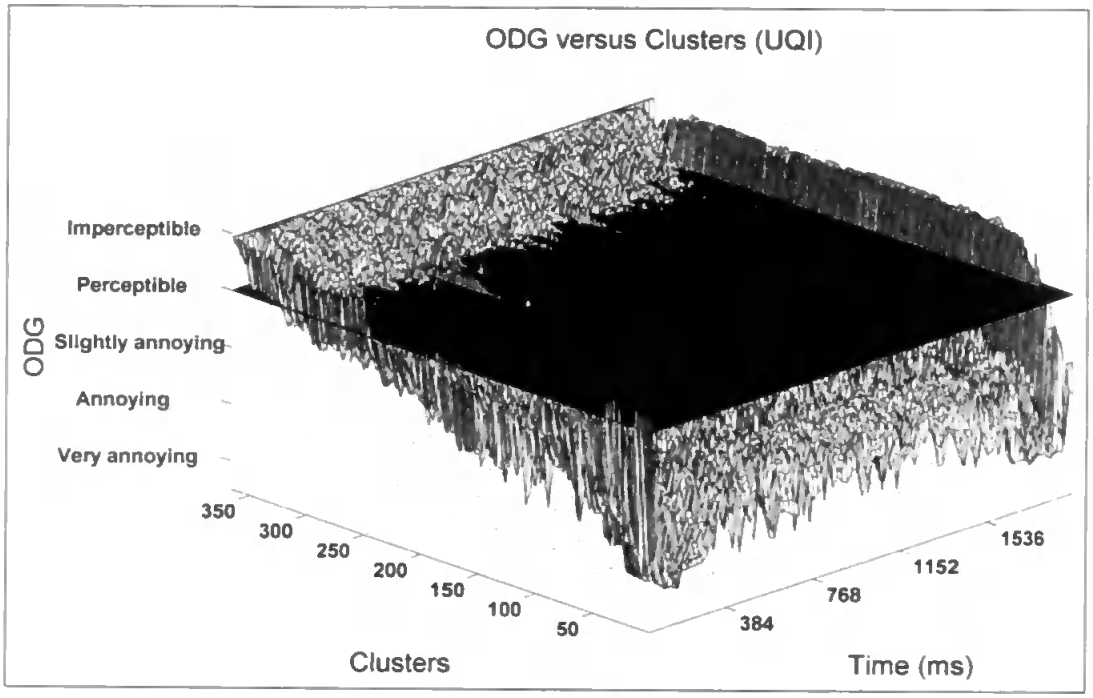


(a) Waveform modeling difference

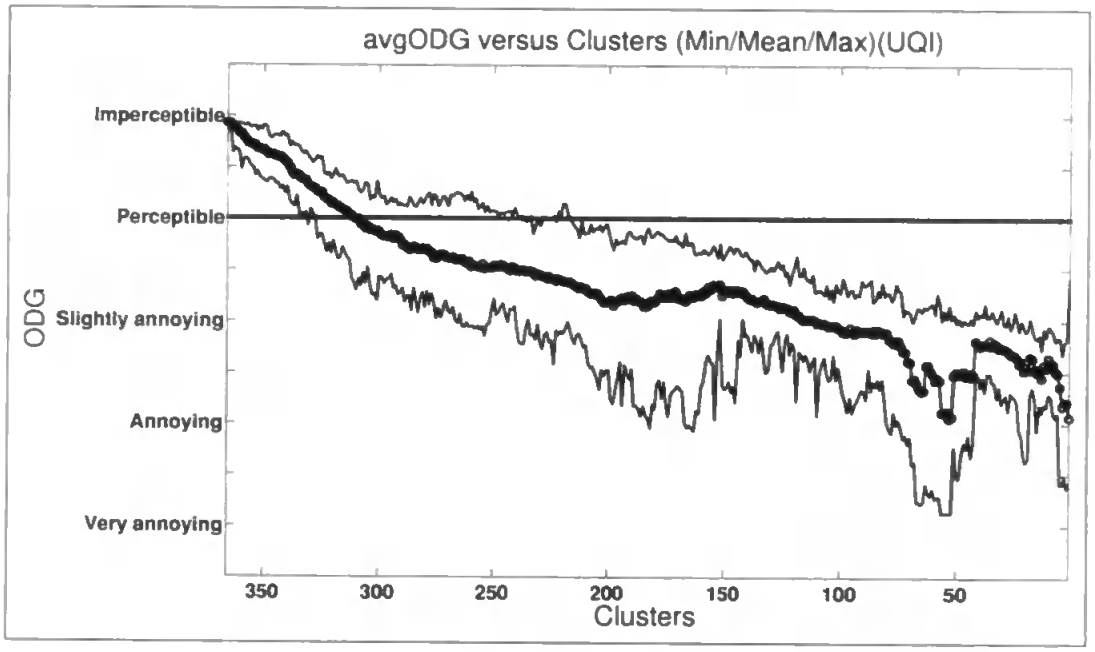


(b) MSE/PSNR Error modeling

Figure 6.64: Trombone - Clustering results (UQI)



(a) ODGvsCL



(b) avgODGvsCL

Figure 6.65: Trombone - Clustering results ODG (UQI)

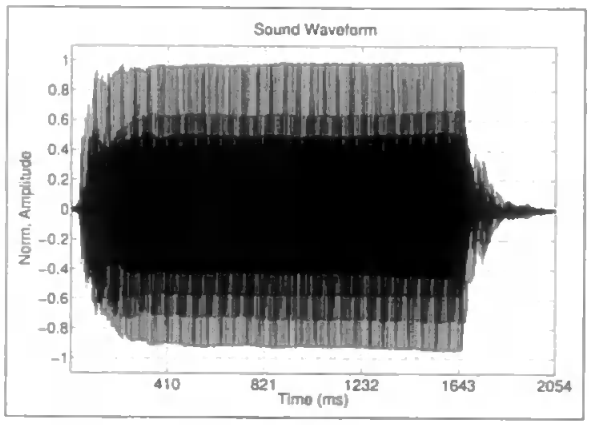
6.4.8 Sound - Oboe 8'

Sound Analysis - This sound is from recent recording by the audio experts of an Oboe 8' pipe. It fundamental frequency was estimate by WinPVoc to be 196.7 Hz. The sound analysis engine results are shown in Figure 6.66(a) for the waveform, Figure 6.66(b) for the spectrum, Figure 6.66(c) and 6.66(d) showing the spectrum lines in linear and logarithmic scales. Figure 6.66(e) shows the harmonics super-imposed with a clearly three most predominant harmonics. However, it can be seen from Figure 6.66(f), showing the harmonic distribution, that interestingly, the highest harmonic is the 6^{th}. The other 1^{st}, 2^{nd}, 3^{rd}, 4^{th} and 5^{th} could thus be considered as sub-fundamentals. There is very little amplitude modulation in this sound.

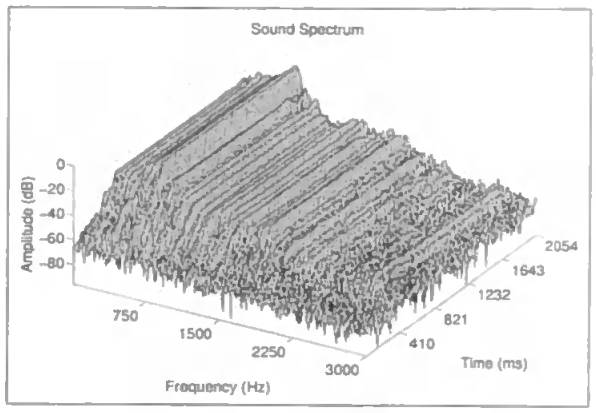
Dendrogram - Figures 6.67(a) to 6.67(l) and Figures 6.68(a) to 6.68(l) show the results of the clustering algorithm with the different methods for calculating the metric distance and linkage methods as described in Section 3.3.5.

Modeling Error - The results for the modeling process the the Oboe sound using the AHC algorithm with 4 different distance metric methods are shown in Figure 6.47 for the CityBlock case, Figure 6.49 for the Euclidean case and Figure 6.53 showing the cases of UQI metric. The results seems to show some important modeling error with the case of UQI having large peaks up to 1.0.

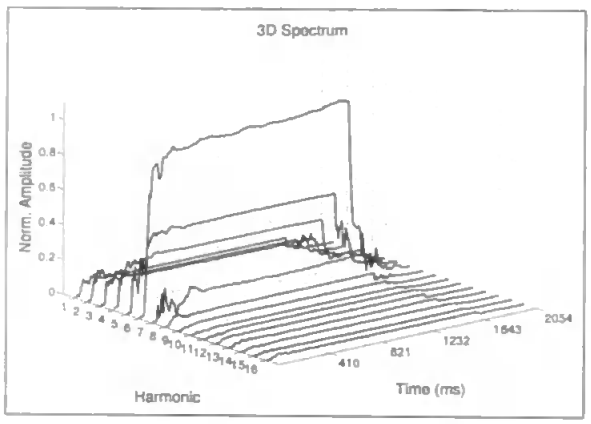
Objective Audio Quality - The results for objective audio quality (surfaces and curves) of the modeling process the Oboe sound are shown in Figure 6.48 for the CityBlock case, Figure 6.49 for the Euclidean case and Figure 6.53 using the UQI metric. Interestingly the objective audio quality curves show a very gradual and smooth audio degradation with a more severe error for the UQI in cases of very small number of clusters. Results indicate that quite a substantial number of clusters can be reduce before the audio degradation would be perceptible.



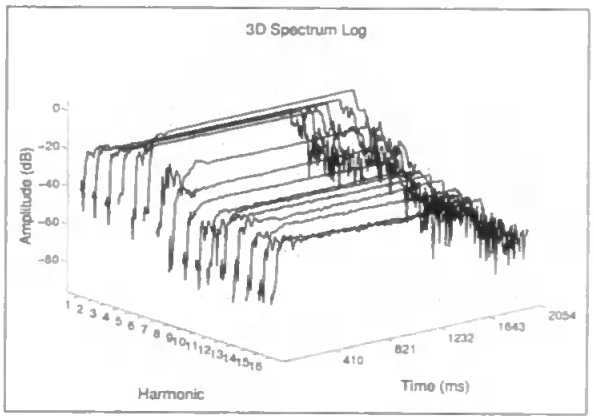
(a) Waveform



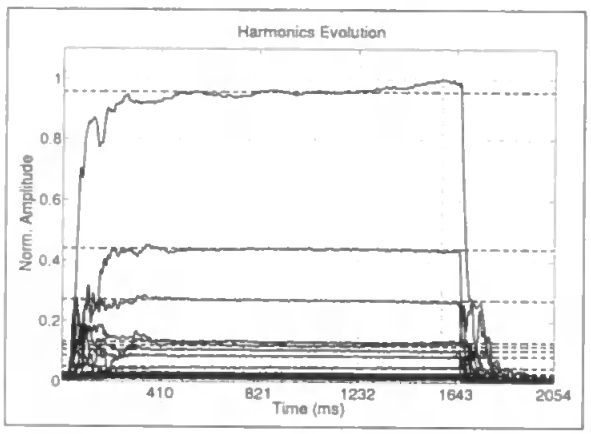
(b) Spectrum



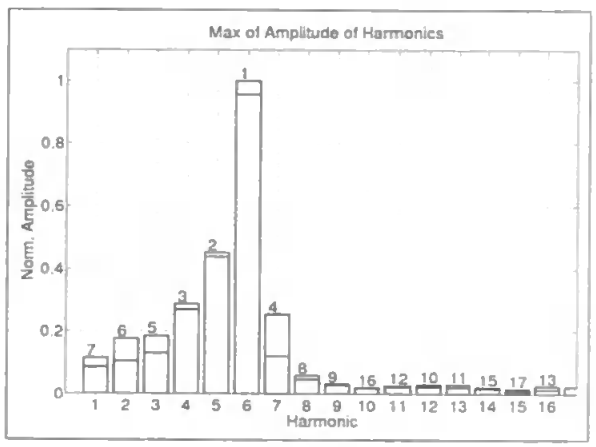
(c) Spectrum lines (Lin)



(d) Spectrum lines (Log)

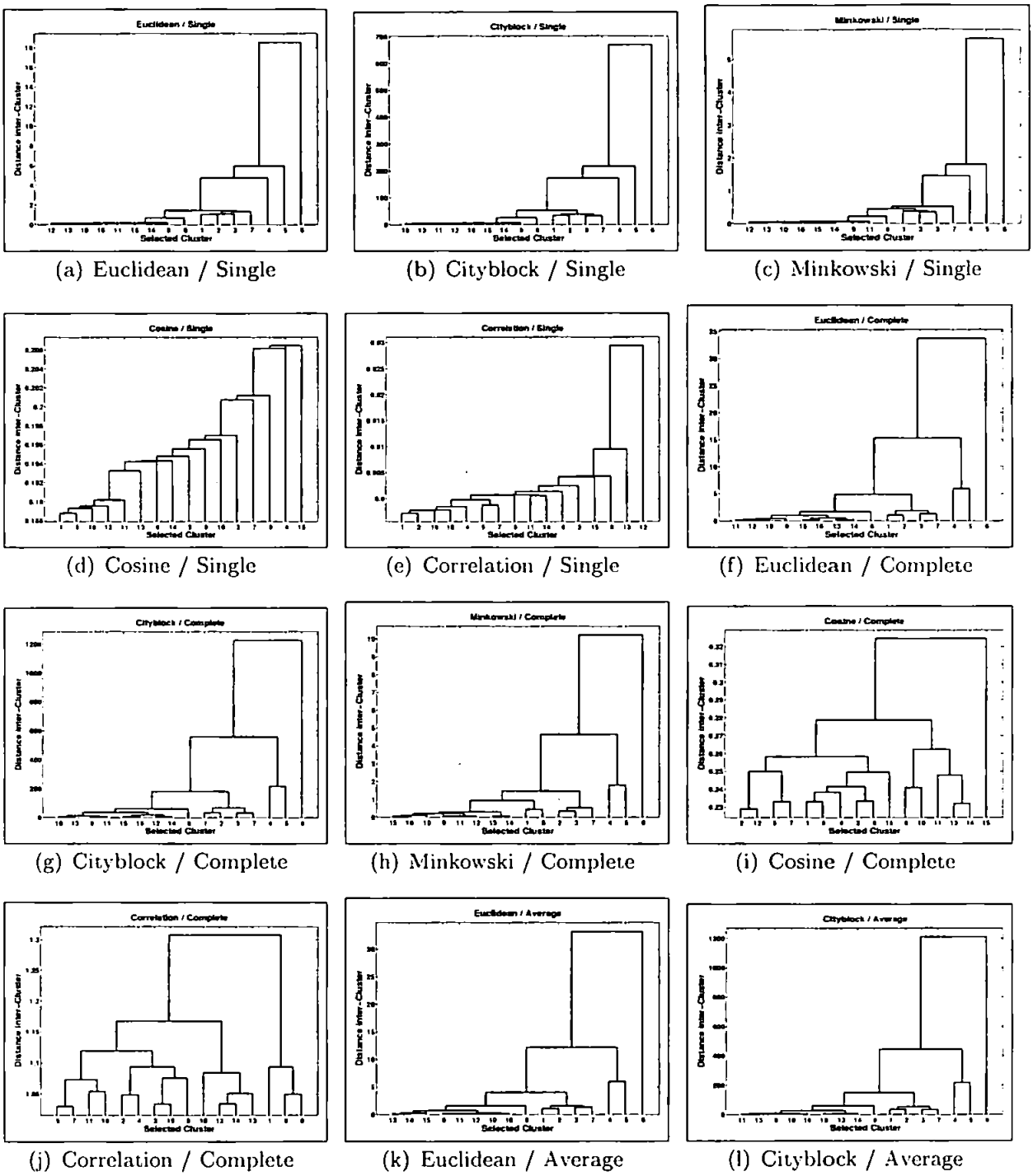


(e) Harmonics envelopes



(f) Harmonics distribution

Figure 6.66: Oboe - Sound analysis



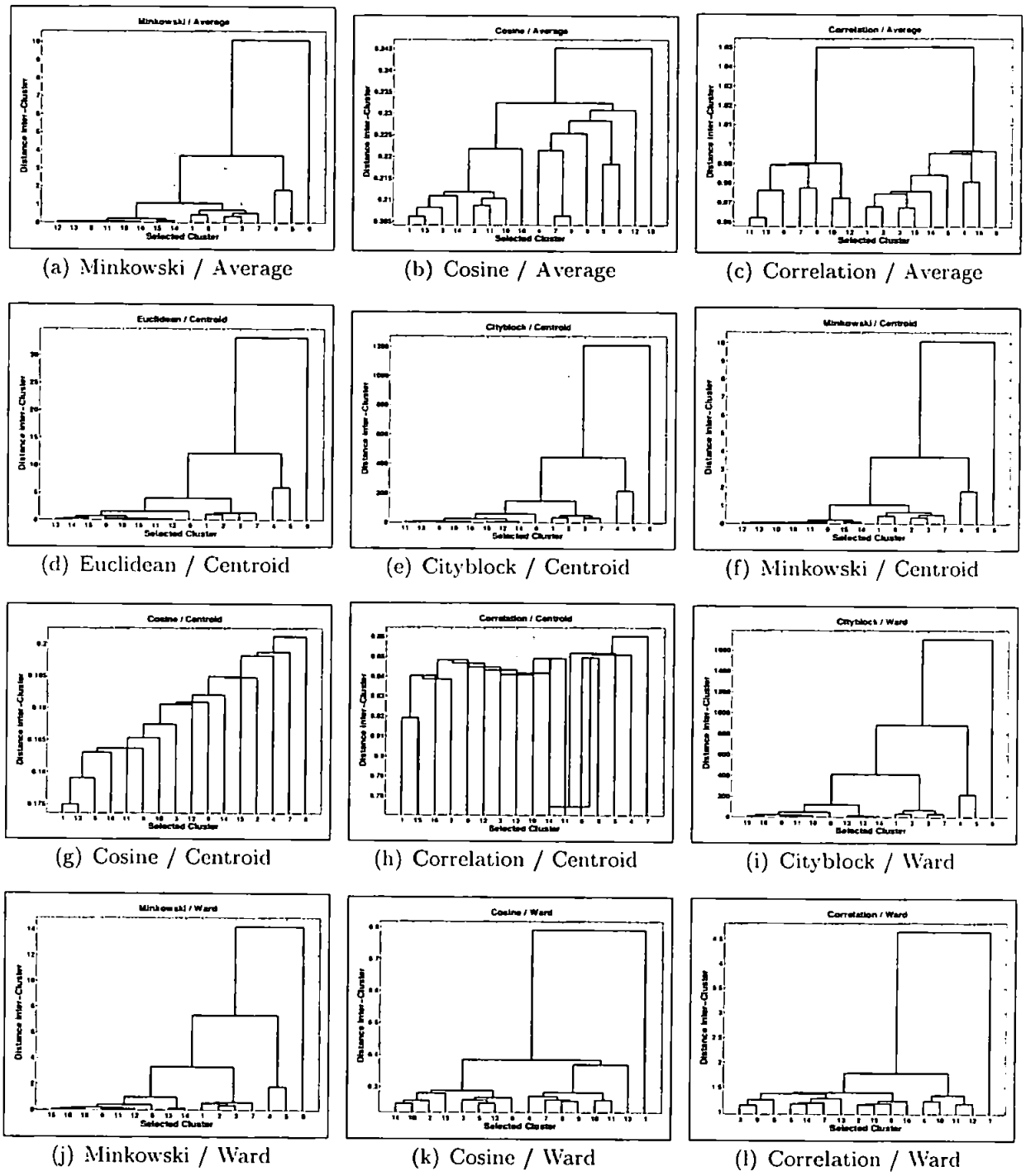
(a) Euclidean / Single (b) Cityblock / Single (c) Minkowski / Single

(d) Cosine / Single (e) Correlation / Single (f) Euclidean / Complete

(g) Cityblock / Complete (h) Minkowski / Complete (i) Cosine / Complete

(j) Correlation / Complete (k) Euclidean / Average (l) Cityblock / Average

Figure 6.67: Oboe - Dendrogram results



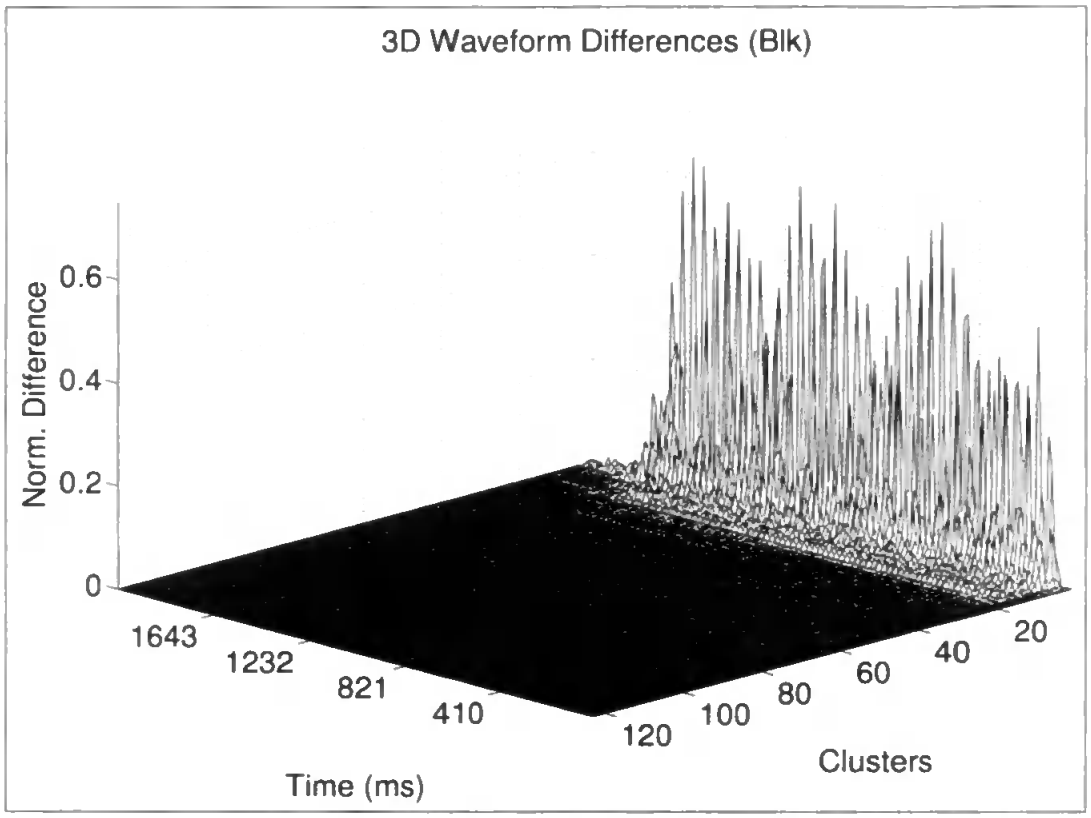
(a) Minkowski / Average (b) Cosine / Average (c) Correlation / Average

(d) Euclidean / Centroid (e) Cityblock / Centroid (f) Minkowski / Centroid

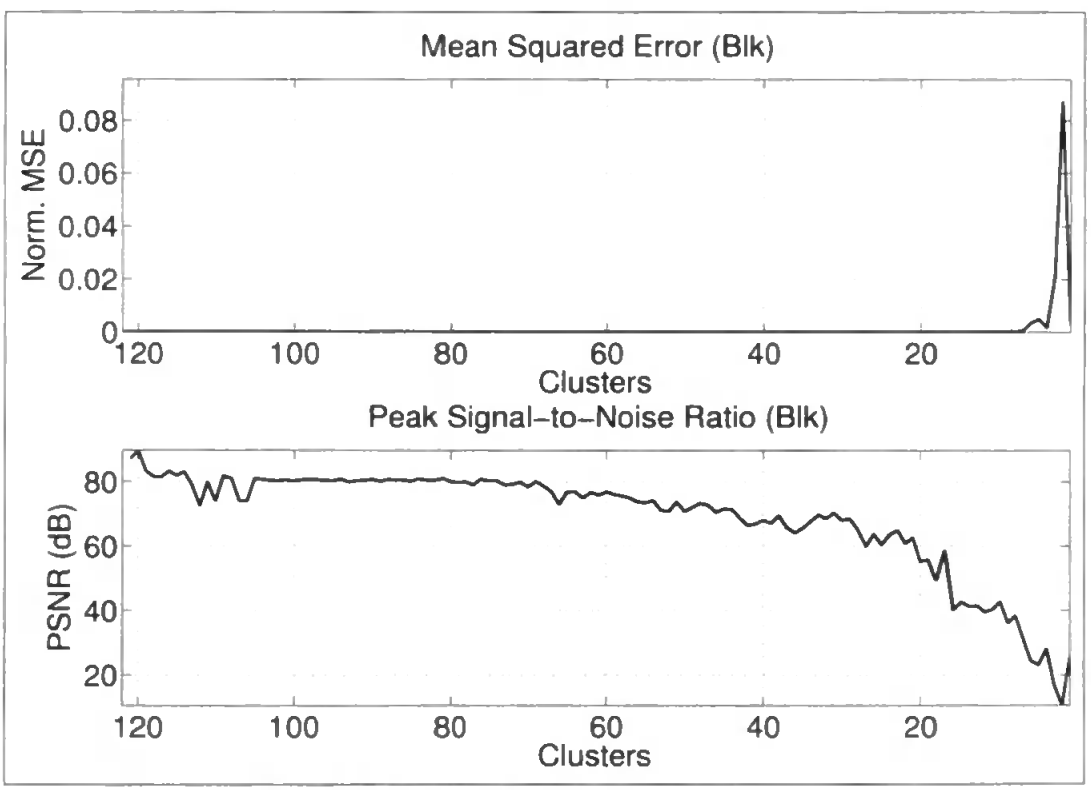
(g) Cosine / Centroid (h) Correlation / Centroid (i) Cityblock / Ward

(j) Minkowski / Ward (k) Cosine / Ward (l) Correlation / Ward

Figure 6.68: Oboe - Dendrogram results (*continued*)

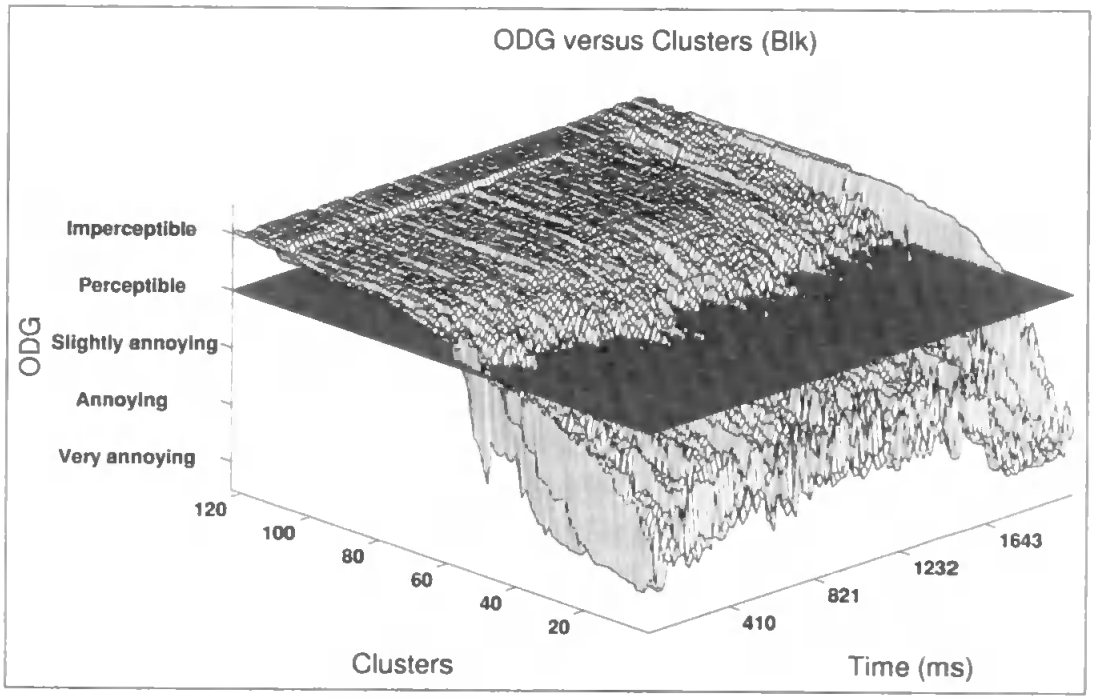


(a) Waveform modeling difference

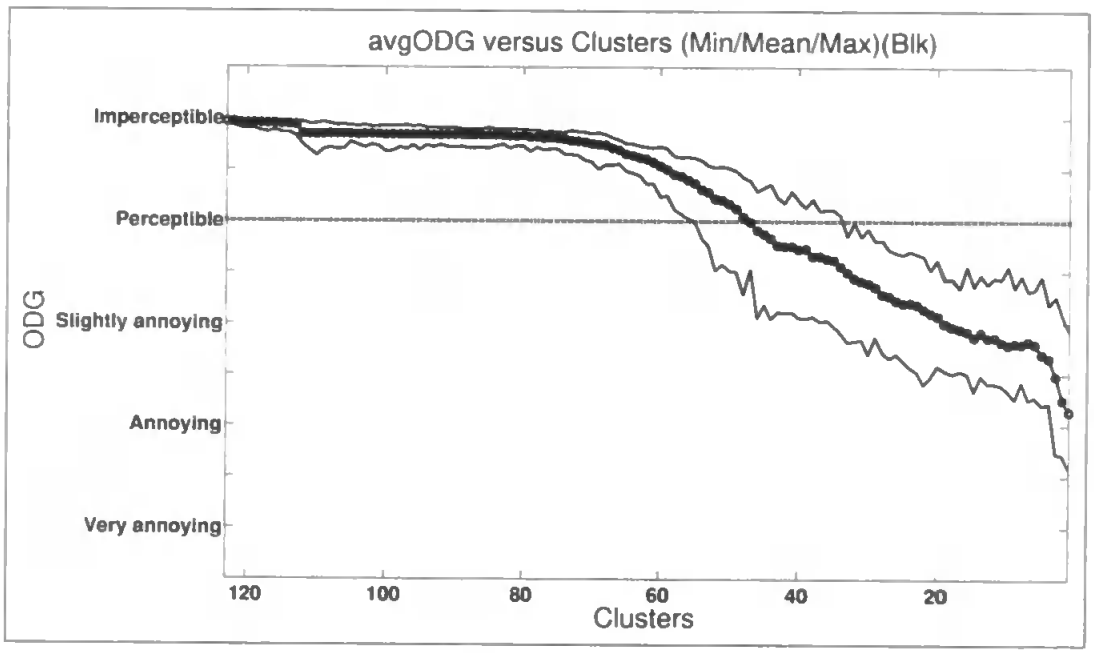


(b) MSE/PSNR Error modeling

Figure 6.69: Oboe - Clustering results (Blk)

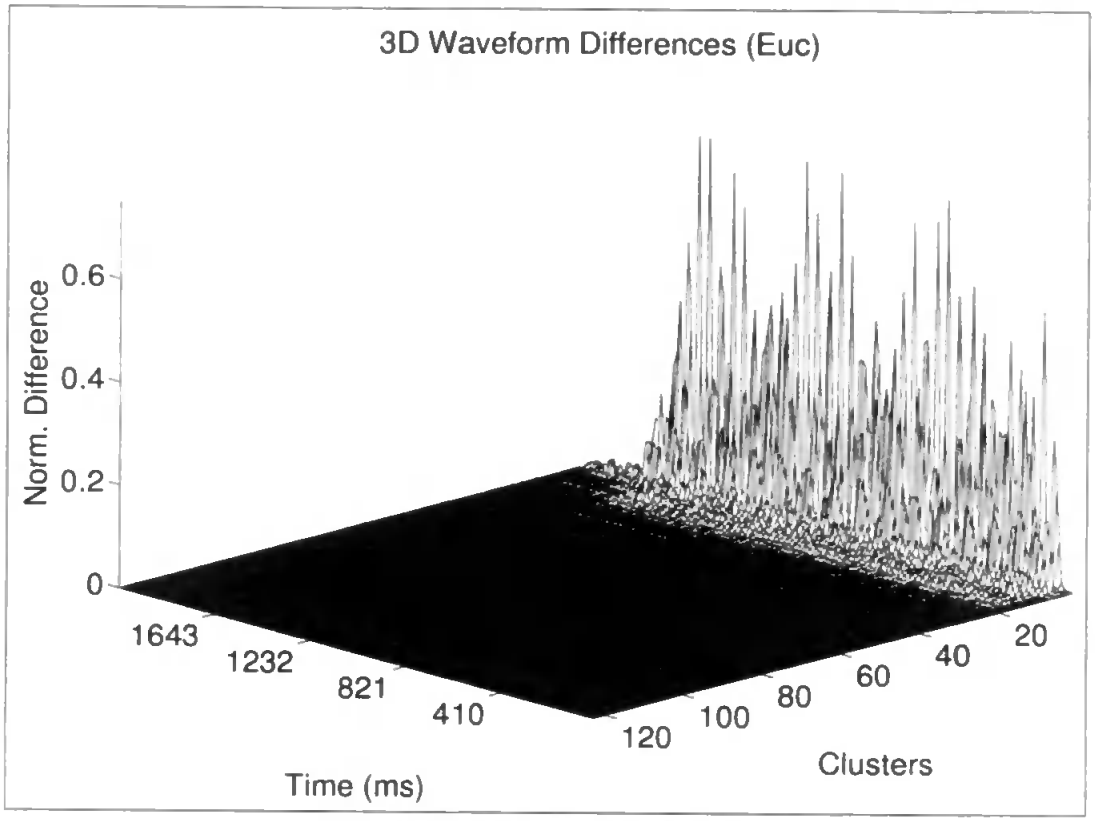


(a) ODGvsCL

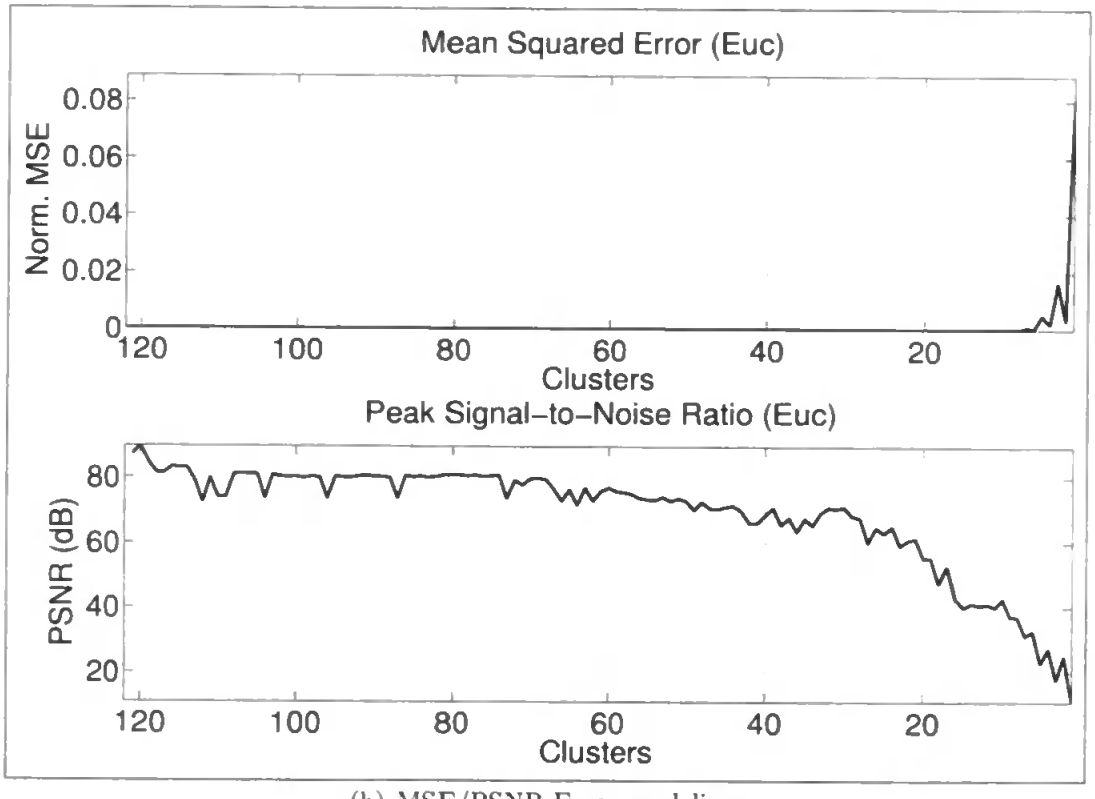


(b) avgODGvsCL

Figure 6.70: Oboe - Clustering results ODG (Blk)

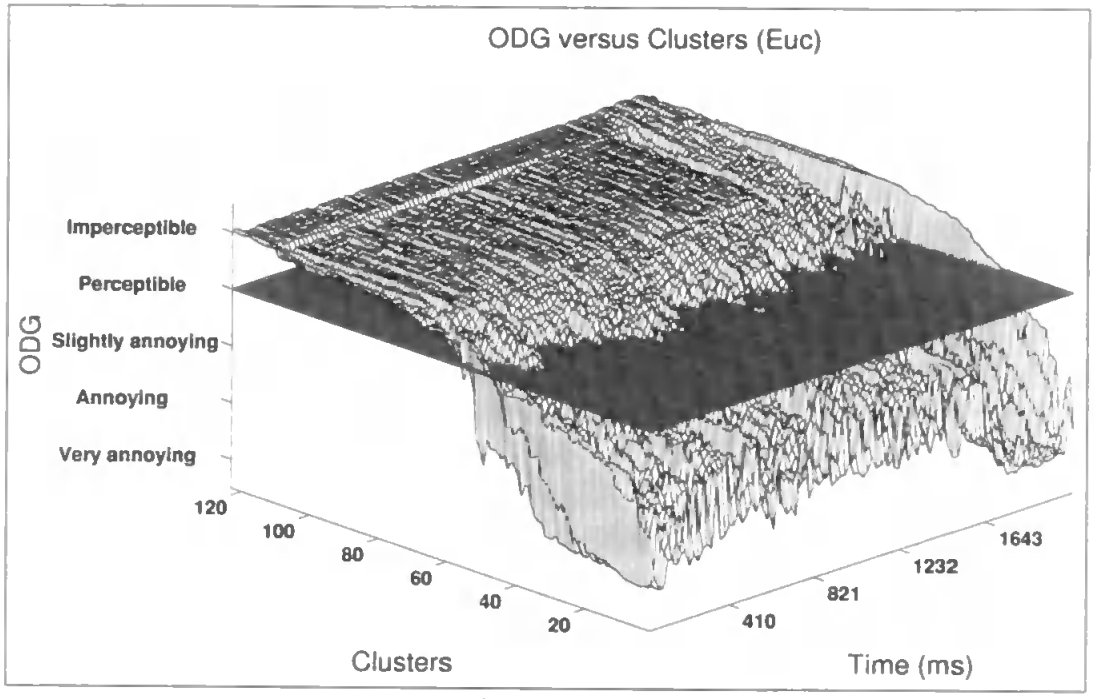


(a) Waveform modeling difference

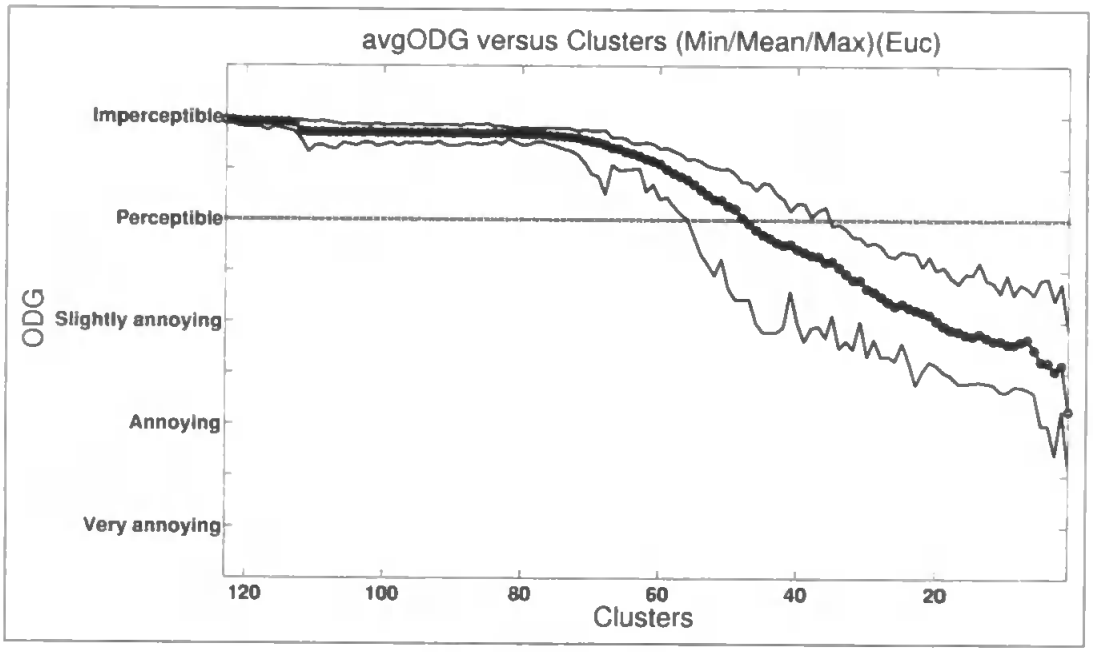


(b) MSE/PSNR Error modeling

Figure 6.71: Oboe - Clustering results (Euc)

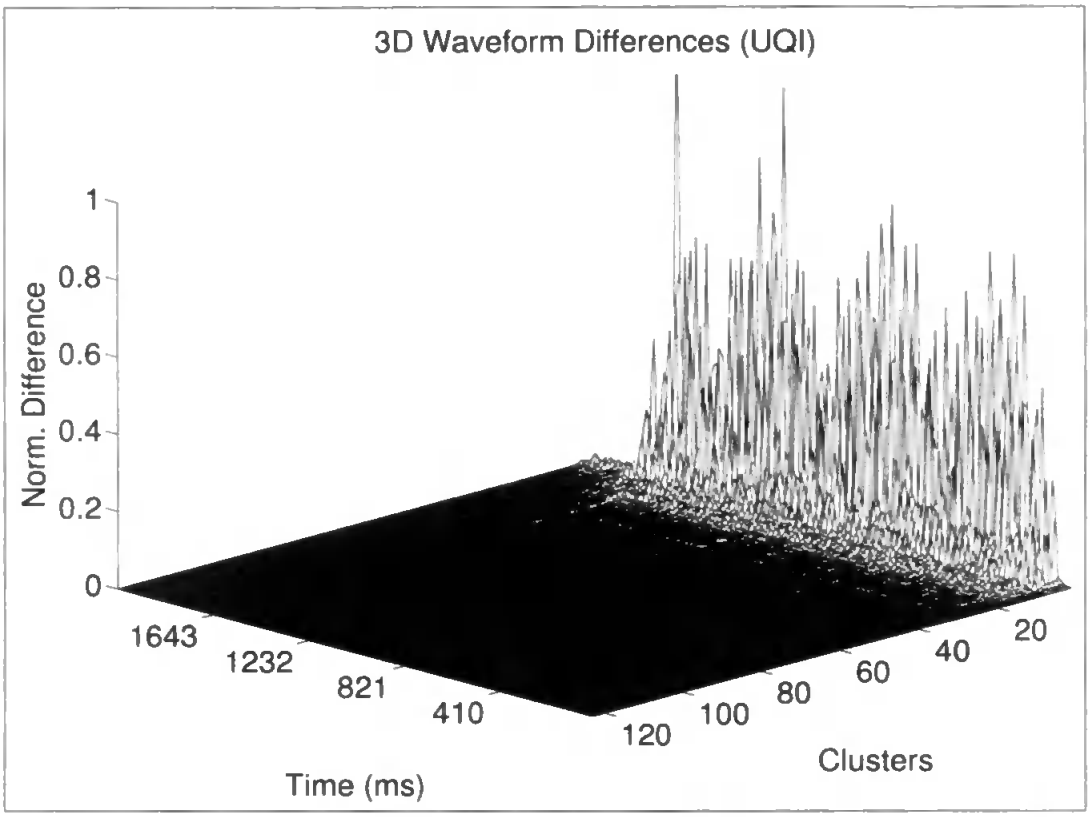


(a) ODGvsCL

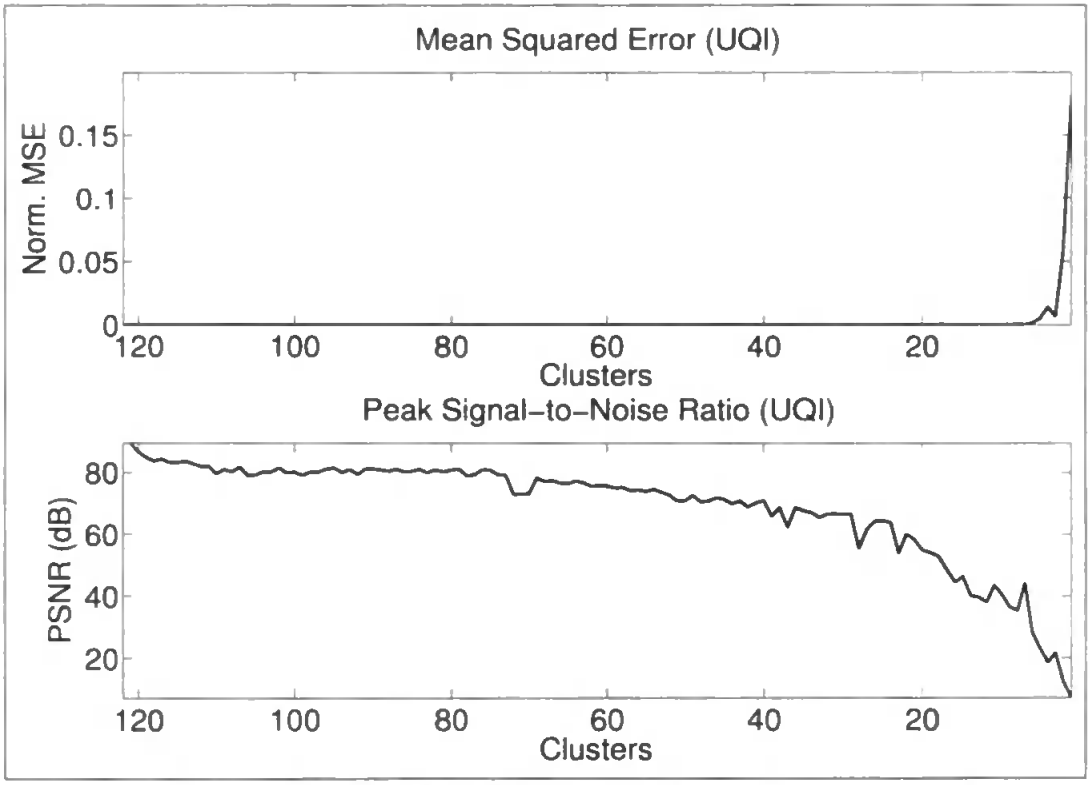


(b) avgODGvsCL

Figure 6.72: Oboe - Clustering results ODG (Euc)

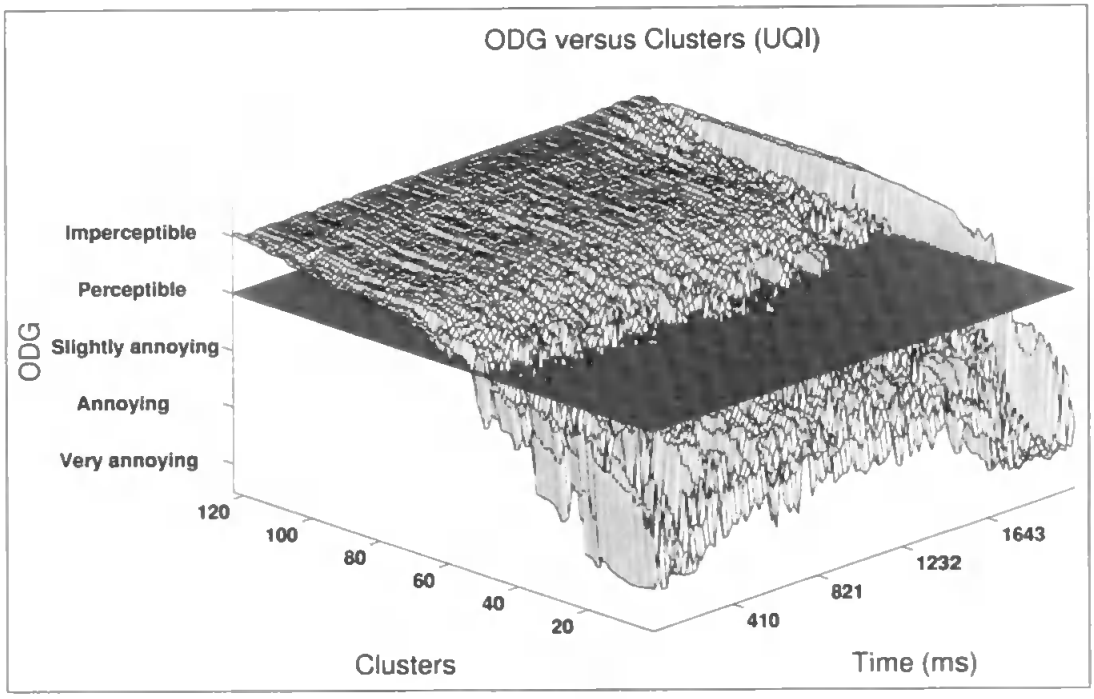


(a) Waveform modeling difference

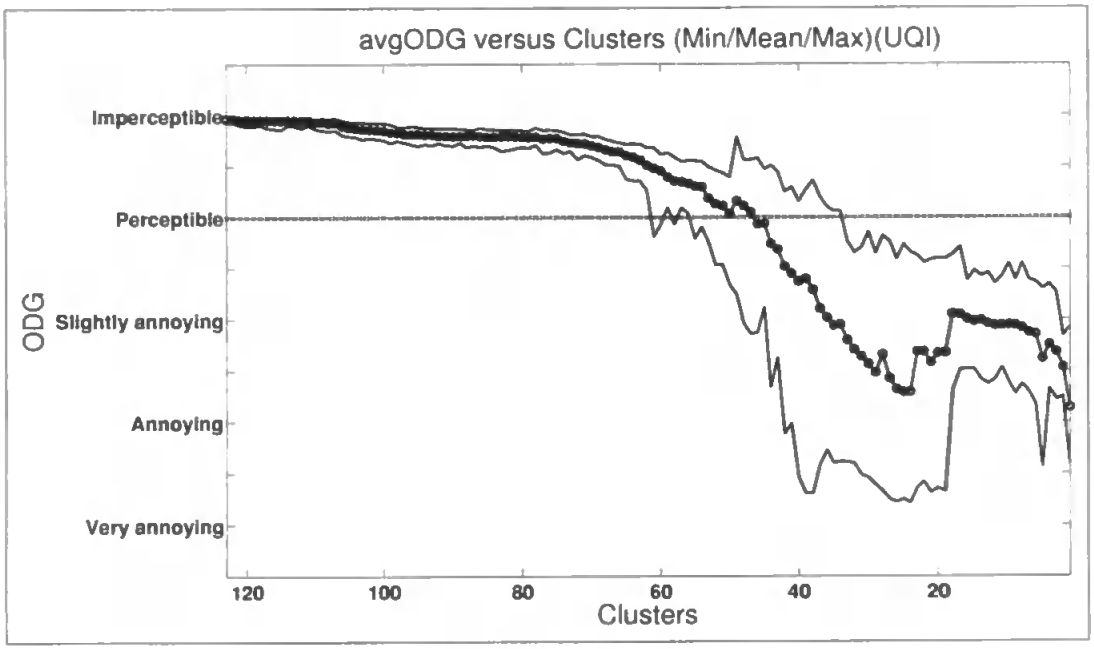


(b) MSE/PSNR Error modeling

Figure 6.73: Oboe - Clustering results (UQI)



(a) ODGvsCL



(b) avgODGvsCL

Figure 6.74: Oboe - Clustering results ODG (UQI)

6.5 Results - Fuzzy Model

A reduced set of experiments have been carried out using the current fuzzy model of audio expertise to design sounds. The evaluation has been limited to the same four sounds used in the objective prediction of sound synthesis quality [61]. The following points can be made:

- The fuzzy model is able to find rather accurately harmonic clusters and define for each harmonic its most important contribution toward the final sound (in terms of attack, sustain and noise characteristics).

- The fuzzy model is faster than the clustering techniques, the reason being that by its design it efficiently emulates the sound design process. The timing functions added to WinPVoc showed that full modeling of sounds such as the Horn example (including the analysis, re-synthesis of all the 219 clustered sounds and assessment of their quality) can take up to 10 minutes. Using four different distance metric methods for each sound, the generation of audio quality surface, like Figure 6.70 to 6.74 for example, can be very time-consuming. An automated sound analysis helps greatly for the process of a full keyboard with keys usually from *C-0* to *C-7* (using file 12.WAV to 96.WAV, see Table A.1).

- The use of the intelligent audio system greatly reduced the time taken to design organ pipe sounds to a quality that is considered high by two audio experts and the fuzzy model of expertise will further reduce this.

- The fuzzy model can allow generation of certain sounds of audio quality often indistinguishable from the original one. Flute sounds, with their simpler harmonic structure, have been found to be best candidate for this preliminary evaluation.

- The audio experts stated [97] "Our system presently requires the skill base of two people - there is only two in the World who can work with this system. In order

to expand the sound 'library' and offer the product to a wider market place, we need to develop more accurate 'semi-automatic' analysis and re-synthesis tools in software". The work presented in this thesis with the fuzzy model provides an efficient solution to Musicom Ltd.

- The development and evaluation of the fuzzy model is very time-demanding for the busy audio experts however, it is believed that future benefits for Musicom Ltd will be great.

6.6 Discussion

In this section, the significance of the results from a research point of view and the benefits they give to the audio experts of Musicom Ltd are discussed. Important points related to the results are detailed and divided into the sound analysis, clustering tree, the modeling error, and the perceptual quality surface.

6.6.1 Sound Analysis

In this research project, techniques have been developed and implemented to improve the sound analysis of original WinPVoc tool. This allowed the extraction of the audio features used by the fuzzy model of audio expertise. Important points are that:

- The sound analysis engine now provides the audio experts with a more detailed sound analysis than originally available by WinPVoc. The results presented now from the Matlab scripts put in evidence key audio features of the sound analysis, and used by the fuzzy model.

- The harmonic distribution for example, showing the distribution along the harmonic axis, was found very helpful for the author. After listening to a sound, the audio experts seem to build a *mental picture* of the sound's harmonic structure and can

tell easily which harmonic has modulations and particularities. This ability has been developed with years of experience performing listening tests of pipe organ sounds, and for the author only the tools can provide such accurate information.

- The attack/decay timing technique gives good estimates of the attack time of the harmonics, crucial information for the sound design as for example the distance metric used in the clustering process does not take into account these perceptual audio features and synthetic sounds for example lose their "percussive" character.

- The perceptual masking (See Section 3.3.3) gives the audio expert some indications about harmonics being potentially masked by others (as seen from Figure 3.7(e)). This further optimises the use of the synthesis resources as being masked, from a perceptual point of view, they should not be required in the synthesis process.

- To improve further the sound analysis engine, high-dimensional visualizations techniques [54] are currently being investigated. The aim is to present combined audio features to the audio experts in ways that they can easily judge, comparatively, the influence of certain audio features (e.g. impact of the amplitude versus impact of the frequency on the final sound quality). The evolution of each harmonic in 3D (with the amplitude vertically and the frequency horizontally) should show trajectories characteristic to certain type of sounds. The amplitude and frequency modulations can also reveal clues to help improve the clustering process, rules of the fuzzy model, future Musicom sound synthesis engine and finally "voicing" of electronic pipe organs.

6.6.2 Dendrograms

This section discuss the results from the dendrogram figures from the results on all the sounds used to evaluate the intelligent audio system. The idea was to provide audio experts with a enhanced and detailed view of the clustering process compared to ordinary

error curve (See Figure 3.11) previously used in WinPVoc tool. Other important points are the following:

- As shown in the example of the previous sections, the dendrograms have very different shapes. This shows that the choice of the distance metric and linkage methods is important as it will affect harmonic clustering aggregation and thus the final sound quality. Two shapes are most common to all these figures: 1) a *rising shape* (like in Figure 6.3(d) and even more pronounced in the case of the flute sound in Figure 6.23(a)) and 2) a *spread shape* (See Figure 6.3(j), and 6.4(c)).

- The original WinPVoc tool implemented a Agglomerative Hierarchical Clustering (AHC) algorithm with an Euclidean distance metric and single linkage method. All the dendrograms show that other alternative methods such as complete, average, centroid or ward have shown to provide other clusters of harmonics and thus, affecting the final sound quality in different ways.

- The chosen sound examples have very different harmonic structure (sounds like flutes with one predominant harmonic and has a much simpler harmonic structure than Basson or Gamba sounds for example), and there is no "best" clustering distance metric or linkage method as such. Only the assessment of the final audio quality allow to determine which distance metric and linkage method are more suitable for specific type of sounds.

- The audio experts often "think and talk in terms of harmonics", the dendrograms show clearly which harmonics are clustered together. Modifications to the original MATLAB functions, with the idea of rearranging the order of the harmonics, improve the visual representation of the harmonic relationships in the clustering process. These developments of *ordered dendrograms* are detailed in [56].

- This research it has been found that there is a firm need develop a perceptual

similarity index specially developed taking into account aspects of sound design. The distance metrics used in this work do not take into account any perceptual issues related to frequency modulation for example. Note that from UQI mathematical definition (See Equation 3.6) it is the only metric really taking into account amplitude variations (based on correlation and distortion).

- The study of the sound's audio features also show that there is a need to develop a "dual amplitude/frequency" perceptual clustering algorithm that would not only operate the basis of the amplitude but also look at the frequency behavior of the harmonics.

- The diversity of the results shows that from the original (Euclidean) distance metric used by WinPVoc, the use of other distance metrics methods can improve slightly the performance of the AHC algorithm. This can be seen from perceptual quality surface figures. The fuzzy model, with its rules explicitly defined for sound design, matches better the requirements for this task than any general clustering algorithms. Investigations into the use of more advanced clustering algorithms such as K-Means Clustering (KMC) [49], the Linde-Buzo-Gray K-Means Clustering algorithm (KBG-KMC), and Fuzzy C-Means Clustering (FCMC) have been recently carried out with preliminary results reported in [56].

- The Preferences DialogBox (See figure 3.16) in WinPVoc allows to select one linkage method and one particular metric from Table 3.2. A new clustering algorithm, in which the most appropriate distance metric and linkage method would be selected at each iteration, depending on the number of cluster left and the re-assessment of the harmonic structure of the sound, could improve the clustering performance. Another idea it to have a multi-pass algorithm, in which in each pass selects a distance metric or/and linkage method that improves to the final sound quality.

6.6.3 Modeling Error

In this section, the modeling errors calculated, for all the sounds, by the Mean Squared Error (MSE)/PSNR and WaveDiff are discussed. The main point made here is these methods do not provide effective ways to assess the real audio degradation of a clustering algorithm.

- MSE and PSNR are very poor in providing any information about the real audio quality (or degradation in the case of a clustering process). Most of MSE Figures show a very flat portion with only a raising (right part for the last few clusters) at the end of the clustering process (only few clusters).

- PSNR figures gives slightly more information than MSE, as it calculation uses logarithm operation, it expand the very small values of MSE. Both MSE and PSNR are calculated for the whole sound and do not provide any timing information.

- WaveDiff results share similarities with MSE results in the sense that they often present a flat portion at the beginning and large error at the end of the clustering process. Clustering examples with the Max distance metric method present more severe clustering errors with even cases at the beginning of the clustering process. Using the log-file feature of WinPVoc, these errors can be detected and their causes closely examined.

6.6.4 Objective Audio Quality

Objective audio quality results are presented in two forms. First, an ODG surface showing PEAQ ODG values (Z-axis) against the decreasing number of clusters used in the synthesis (X-axis) against time (Y-axis). Figure 6.8(a) and 6.37(a) are examples of these objective audio quality surfaces. Secondly, results are shown as ODG curves with minimum, mean (averaged over time) and maximum values, as shown in Figure 6.32(b)

and 6.54(b). For each sound example, results are given for 4 different distance metric methods (Blk, Euc, Max and UQI). From the study of these figures, the main points are the following:

- The audio quality surface figures offer potential to have an overview of the whole clustering process. The motivation was to provide the audio experts with clear visual representation of the degradation of audio quality due to the clustering process.

- The audio quality surfaces shown are *true* representation of the objective audio quality resulting from the clustering process, calculated using the PEAQ algorithm. They clearly show the impact of decreasing number of clusters onto the objective audio quality. The reference sounds used for these plots are the best synthetic (i.e. sound synthesised using full data of the Phase Vocoder analysis) and each step represents the PEAQ calculated audio degradation as the number of clusters reduces.

- The *perceptible plan* or *perceptible line* (on both figures where values of ODG equal to -1) has been added to the plots to help estimate by how many clusters the algorithm can reduce the sound design process before the sound degradation becomes perceptible. It is worth noticing that this perceptible threshold does not always correlate with the audio experts' judgement as to whether the tones are still considered of high quality.

- Some quality surface results present distinct variations. Figure 6.41(b), 6.52(b) and 6.63(b) show that the Max metric produce substantial drops in the audio quality, whereas in the majority of cases the UQI metric gives smoother results than using Euclidean (Euc) and Cityblock (Blk). This provides the audio experts with indicators about the distance metric method and clustering algorithm performance and help to detect bugs and modeling artefacts, and most importantly at what stage they occur (i.e. iteration step), and examine them using WinPVoc's log-file.

- The perceptual surface quality also provides information about timing in the modeling process. If Figure 6.32(a) and more apparent in Figure 6.50(a) are compared, we can see the quality surface showing a drop of quality at the end of the attack segment. The same applies at the release portion (which are often attributed to sudden change of SNR for which PEAQ is very sensitive).

6.7 Summary

In this chapter, the evaluation of the intelligent audio system was presented. A sound database was created from 4 CDROMs of pipe organ recordings provided by Musicom Ltd. MATLAB scripts have been developed to automate the sound analysis task of this large database and thus reduce the work-load of the audio experts. Sound examples have been chosen from the database which included a Basson 8', Flute, Horn and Principal 4' with also more recent recordings of a Dulciana 16', a Metal Trombone 16' and Oboe 8'. All these sounds have very different harmonic structures, and are ideal to evaluate the intelligent audio system performance. Analysis results were presented with the sound analysis, dendrograms and MSE/PSNR plots of the modeling error and finally, perceptual quality surfaces and curves calculated using the PEAQ algorithm. A limited number of experiments have been carried out to evaluate the fuzzy model of audio expertise described in Chapter 5. At the time of writing, the fuzzy model is still under evaluation by the audio experts and continuous development by the author.

Results presented in this chapter have been published in a conference paper at the 115^{th} Audio Engineering Society convention in New York in October 2003. A copy of the Preprint is in given in Appendix C.3 and more results are also available from [36].

Chapter 7

Discussion, Further Work and Conclusions

7.1 Introduction

The aim of the project on which this thesis is based is to develop a new Artificial Intelligence-based method to aid modeling of musical instruments and sound design. This final chapter gives a summary of the research carried out, discusses the main contributions to knowledge, identifies the limitations of the research, proposes directions for future work, and finally concludes the thesis.

7.2 Summary of the research carried out

The initial work focused on setting up and establishing real-time audio research platforms using a DSP card and a multimedia computer. The main goal was to assess the effect of the model parameters on the resulting timbre and perceived sound quality. A MIDI interface was built to connected a MIDI Wind controller and MIDI keyboard to the DSP card. Sound synthesis models were coded in DSP assembler language for the Motorola DSP56303 and optimised by hand for speed and memory limitations. A DSPShell tool was also developed to facilitate the DSP development in assembler language. Both real-time audio platforms can sustain high quality real-time sound synthesis and can also be

used for sound analysis and processing as well. Most common types of sound synthesis including additive synthesis, frequency modulation, wavetable synthesis and digital waveguide modeling have been developed and implemented using Microsoft Visual C++ 6.0 (C++/MFC) as individual tools (AddSynth, FMSynth, etc.). To facilitate the evaluation of each sound synthesis techniques, GUI control elements (slider, button, etc.) allow to control all the sound model parameters and give freedom to explore in real-time the true potential of each sound synthesis. This work was presented in a conference paper [59] at the DAFX-99 conference in Trondheim, Norway.

A sound database has been created from a set of 4 sound banks on CDROM provided by Musicom Ltd. The sound database, described in Section 6.2, contains pipe organ sounds collected during recording sessions at organ builders and Musicom customers (church and cathedrals) in European and mostly in United States. The database is nearly 2 GBytes of sound data, and thus MATLAB scripts have been developed to automate sound analysis [57] and help to detect particular sounds suffering from background noise (See Section 3.2.1).

An intelligent audio system has been developed, which consist of four main parts: a sound analysis engine, audio feature processing engine, a sound analysis engine and a sound quality assessment engine. This work was presented in Chapter 3. The intelligent audio system was implemented in C++ and MATLAB scripts, and integrated into the WinPVoc tool (originally developed by Musicom Ltd).

Results from the sound analysis engine, based on Phase Vocoder, provided time-varying spectro-temporal characteristics of the sound, from which audio features have been and extracted. These audio features have been defined and discussed with the audio experts and represent key aspects of the sound analysis considered by the audio experts in the design of pipe organ sounds. Chapter 4 described the sound analysis and audio features extraction with illustrative examples. These audio features are used as inputs to the audio feature processing engine.

The audio feature processing engine is a fuzzy expert system that implements the *fuzzy model of audio expertise*. This computational model, based on fuzzy logic techniques, emulates the decision-making process used by the audio expert in the complex and time-consuming task of sound design. It represents one of the main contributions of this thesis. The design and development of the fuzzy model of audio expertise was presented Chapter 5. Knowledge elicitation session (including formal and informal interviews, email discussions, etc.) helped to capture, for the first time in computer music, the audio expertise required for sound design and for developing the fuzzy sets, variables and rules of the fuzzy model.

The fuzzy model was implemented using two software solutions: first within the MATLAB environment using the Fuzzy Logic Toolbox to allow rapid prototyping and evaluation and secondly, a fuzzy library and a fuzzy system have been developed in C++/MFC for future integration into WinPVoc. This work was presented in Section 5.10.

The intelligent audio system with the fuzzy model and preliminary results were first presented at the 111^{th} Audio Engineering Society convention in New York, December 2001, with a paper [59] that attracted significant interest from audio researchers. To the author's knowledge, no equivalent approach has ever been published and this proves the innovative aspect the work presented in this thesis.

The sound quality assessment engine, which is the final part of the intelligent audio system, is used for the quality assessment of the synthetic sound produced by system. It is based on the ITU-R BS1387 standard, the Perceptual Evaluation of Audio Quality (PEAQ) algorithm. The sound quality assessment engine has been developed to use the Opera system, the only available commercial solution of the PEAQ algorithm, with MATLAB scripts that automate the sound quality assessment and visual representation of the Objective Difference Grade (ODG) results of the sound produced by the intelligent audio system. This work was published as a conference paper [59] at the 115^{th} Audio Engineering Society convention in October 2003 and is the first publication on objective

prediction of sound synthesis quality.

Experiments have been carried out on reduced set of sounds covering a broad variety of pipe organ sound characteristics. The results were presented in Chapter 6 including discussions about their significance and benefits to audio experts. More results on the large database are summarised in [57].

7.3 Contributions to Knowledge

The most significant contributions of this work are: real-time sound synthesis, intelligent sound design and objective prediction of sound synthesis quality.

7.3.1 Real-time Sound Synthesis

The first achievement of this work is concerned with the development of real-time sound synthesis research platforms. Specific achievements are:

- **Development of real-time sound synthesis platform.** DSP-based and computer-based development of audio research platforms for real-time sound synthesis (and potentially sound analysis and processing) have been established. This allows us to evaluate the current state-of-the-art of modern DSP technologies for future electronic pipe organ sound synthesis systems update.

- **Development of a real-time sound synthesis.** Software tools have been developed, implemented on both DSP and computer platform, that implement in real-time, additive, frequency modulation, wavetable and digital waveguide sound synthesis techniques. This also allows us to estimate DSP resource requirements for future electronic pipe organs sound synthesis.

- **Assessment of sound synthesis parameters on perceived audio quality.** Software tools have been developed allowing the audio experts to assess, in real-

time, the effect of sound model parameters on the perceived audio quality to reveal the true potential of each technique with particular interest for pipe organ sound synthesis.

7.3.2 Intelligent Sound Design

A key achievement of this work is the development of an intelligent audio system based on computational intelligence and auditory perceptual techniques to automate the complex task of sound design and modeling musical instruments. This contribution was published in [60]. Specific achievements are:

- **Development of Fuzzy Logic models of audio expertise for sound design and modeling musical instruments.** For the first time in computer music, a model of audio expertise has been developed using fuzzy logic techniques that can handle imprecision and uncertainty inherent in the audio knowledge and sound data. The model mimics audio experts reasoning using audio features that are considered key in a sound design process of high quality sounds. The rules used in the decision making process have been developed after knowledge elicitation sessions with two music technologist/audio experts.

- **Development of a fuzzy expert system.** New techniques to capture and exploit, for the first time in computer music, the audio expertise of audio experts for sound design and synthesis have been developed and implemented into an intelligent audio system. A prototype fuzzy expert system that capture and exploit this audio expertise has been developed, implemented and evaluated. Pipe organ sound synthesis has been used as a vehicle for the investigations. Limited evaluation of the system showed that it can be used to design simple sounds which the experts judge to be of very good quality, and reduces the work-load of the audio experts.

7.3.3 Objective Prediction of Sound Synthesis Quality

For the first time in computer music, an innovative use of objective method for the prediction of sound synthesis quality is developed. This method has been published in [61] at the 115^{th} Audio engineering Society convention in New York. Specific achievements are:

- **Development of objective prediction of sound synthesis quality** New approach to predict sound synthesis quality using objective method (PEAQ ITU-R BS1387), removing the need for listening tests. For the first time in computer music, the PEAQ algorithm is used to assess the sound quality of a sound synthesis system.

- **Automated sound design system** New sound design system that fully automates the process of sound analysis, audio features extraction and processing and generates optimal sound synthesis parameters with final assessment of objective sound synthesis quality.

- **PEAQ-driven sound design system** Development of novel methodology using objective sound quality assessment technique in the development of sound synthesis and electronic musical instruments. The audio quality assessment is used in the feedback loop and provide indication to control, objectively the sound synthesis parameters, improving the accuracy of the modeling process.

7.4 Limitations of the current work

Clearly the work presented in this thesis is new and has not matured to a state where it could be used in a real-world product. All the software developments have been integrated into WinPVoc and can be used by the audio experts. It is intended that this work will provide the basis from which to develop a commercial intelligent sound design system that

will demonstrate the potential of AI-based techniques for modeling musical instruments. To do this however, a number of limitations exist which will have to be addressed and overcome. These are discussed the point by point.

- **Incomplete validation of the Fuzzy Model** - The fuzzy model has yet to be fully validated by the audio experts and work is carried out by the author to help automating this validation using the large sound database. Furthermore, specific pipe/stop fuzzy model of audio expertise can reduce the need for such large validation, focusing the audio expertise required to develop the fuzzy model to issues only related to the Diapason or flute for example.

- **Complexity of Pipe Organ Modeling** - This project deals with all the parts of the complete modeling of pipe organ sounds, from recording to sound analysis, audio feature extraction and processing, sound synthesis to the final sound quality assessment. Each part requires audio knowledge on its own and the audio experts have it all. They have gained experience over years of practice in recording sessions, analysis of case examples from organ builders, design of the Musicom ADE, and continuously listening for the best quality. As an acoustic instrument the pipe organ has often been named the "king of instruments" and this work has been trying, as much as possible, to gain understanding of the complexity of pipe organ modeling.

- **Suitability of PEAQ for Sound Quality Assessment** - In this work, subjective and objective methods have been used to assess the quality of the pipe organ sounds. Listening tests have been carried out by the audio experts and they are the standard manner to assess audio quality. PEAQ as seen from the results presented in Chapter 6 is very sensitive to small variation of audio degradation. PEAQ has been design primarily to assess the quality of audio codec and its cognitive model developed using audio databases made by audio compression (See Section 2.5.3). PEAQ algorithm suffers greatly from misalignment, i.e. the reference and test sig-

nals have to be perfectly time-aligned to provide accurate results. Using the best version of the synthetic sound as the PEAQ reference signal a simple solution to this limitation. This remains fully valid because 1) the reference sound can be considered as indistinguishable to the original sound (PEAQ curves being flat and showing an imperceptible audio degradation) and 2) the perceptual quality curve and surface represent thus, a true sound quality degradation when the synthesis resources are reduced (i.e. number of clusters used by the sound synthesis engine). From all the experiments carried out, it was concluded that novel methods are needed for sound synthesis quality assessment, and the current PEAQ can provide the basis to do so. Its cognitive model can be re-trained (using the same Neural Network topology) using database of sound synthesis examples, which would be more representative of sound synthesis artefacts. This is the main topic for the future developments of novel sound quality index described in Section 7.5.5.

- **Industry Collaboration** - A limiting factor in this work has been, sometimes, the lack of the accessibility to the audio experts. It must be understood that both Musicom's experts are very busy audio professionals, always travelling for recordings sessions, installing electronic pipe organs systems all over the world. The business/commercial side of such small company takes, understandably, priority over the research and development. However, feedback provided by the audio experts have always been very valuable for the progress of this work. Since 2001 the focus of this project has been put on the following aspects of the project: the fuzzy model of audio expertise and the objective evaluation of audio quality. These aspects with others will extended further as part of the EPSRC project (Grant GR/S00859/01) [36].

7.5 Future Work

Many interesting avenues for future research have been identified and include enhancements of the intelligent audio system and fuzzy model of audio expertise, developments of an advanced sound database, development and implementation of a novel sound synthesis quality index, and finally to use the present AI-based approach for other musical instruments and in other fields of audio research.

Work has already started on most of these topics as part of a fellowship funded by the Engineering and Physical Sciences Research Council (EPSRC Grant Reference: GR/S00859/01) and Master projects proposed and supervised by the author. More information about the EPSRC project can be obtained at the project home page[†] or directly from the author.

7.5.1 Extend the Intelligent Audio System

The intelligent audio system has been developed in such way that each of its four parts can be easily extended in future. The most interesting extensions are on the fuzzy model of audio expertise and advanced clustering techniques.

- **Fuzzy Model extensions** - The fuzzy rules that define the model of audio expertise presented in Chapter 5 have been constantly evolving following series of experiments suggested by with the audio experts. The rules presented in this thesis should be considered as a prototype system that the audio experts are evaluating, with results of preliminary performance presented in Section 6.5. There are many possible extensions regarding the fuzzy model and include investigations into the membership function shapes/positions and their effects on the performance of the intelligent audio system.

[†]`http://www.tech.plymouth.ac.uk/spmc/S00859/`

- **Automated Knowledge Discovery** - The large sound database provide data sets to investigate automated techniques used to evolve the fuzzy model. The using a perceptual audio quality driven optimisation of the fuzzy model optimisation techniques. As suggested in Section 5.9.7, potential development of the current fuzzy model has also been identified for pipe/stop specific models of audio expertise. In these cases, the model will be optimised for one particular type of pipe (like flutes or diapasons for example)using optimisation techniques developed in [43] based on Simulated Annealing (SA) [2] and used to tune fuzzy model of a medical system.

- **Type-2 Fuzzy** - The fuzzy theory used in this research project is of Type-1 fuzzy and recently, a Type-2 fuzzy theory has recently been developed which generated a lot of interest in the fuzzy logic research community. It provides an efficient solution to overcome one major limitation of Type-1 fuzzy theory, which is the vagueness of the fuzzy sets. Important choices have to be made in the design of fuzzy system such as the choice of the fuzzy sets (membership functions) for example. Type-2 fuzzy deals with this limitation by adding some variability into the fuzzy sets. A comprehensive introduction to Type-2 fuzzy sets can be found in [120], with additional reading in [87] and [70]. Ozen and Garibaldi have described recently an investigation into the adaptation in Type-2 Fuzzy Logic systems applied to umbilical acid-Base assessment [125], extending the work from [42]. It is thought that this research direction would lead to a much more accurate fuzzy model of audio expertise.

- **Advanced Clustering Techniques** - Algorithmic solution to the clustering problem of harmonics can be further extended from the basic Agglomerative Hierarchical Clustering (AHC) algorithm implemented in WinPVoc. The topic of data mining and clustering provide a lot of readings and two main issues are *How Many Clusters?* and *Which Clustering Method?*. Initial work has already started using

K-Means Clustering (KMC), the Linde-Buzo-Gray K-Means Clustering algorithm (KBG-KMC), and Fuzzy C-Means Clustering (FCMC), all implemented in MATLAB with preliminary results reported in [56].

7.5.2 Advanced Sound Database

The intelligent audio system works with wave files, i.e. recordings of musical instruments. For this research project, all the data collection (i.e. recording and archiving process) was left to the audio experts who have extensive experience in pipe organ recording techniques. Sound data was supplied in forms of CDROM (See Appendix A.1). However, it has been found that the quality of some of the original recordings was not as good as expected. Some recordings are suffering from substantial background noise and the current intelligent audio system sometimes mis-interpret this background noise as being part of the sound. Furthermore, some pipe organ sounds have very breathy characteristics which makes the modeling task very challenging and has a great influence on the final perceived sound quality of the synthetic sounds and thus on the performance of the system. It was also found that the PEAQ algorithm used in our experiments is also very sensitive to small noise [11], being primarily developed for the assessment of audio codecs and its cognitive model trained using audio database having audio coding artefacts [38]. Noise that, as seen in the development of the fuzzy model, is necessary to improve sound synthesis has a reverse effect when it comes to use PEAQ. All the above problems, help to define future work leading to the development of an advanced sound database.

- **Other Sound Analysis** - More robust techniques such as the McAulay-Quatieri method [118] and Bayesian paradigm [167] will be added to the current Phase Vocoder analysis. Wavelets analysis techniques [?] with the recent complex wavelets [168] are also under consideration.

- **ICA-based Sound Analysis** - Sound analysis techniques based on Independent

Component Analysis (ICA) will be investigated. This is based on the idea that the harmonics of a sound could be seen as independent signal sources for which ICA provides an efficient method to separate. Recent results reported in [55] suggest that ICA has potentials as a sound analysis technique.

- **Fuzzy Logic Signal Processing** - To tackle the problem of noise reduction in pipe organ sounds, novel Digital Signal Processing (DSP) techniques will be developed inspired from Fuzzy signal Processing concepts described in [141] and [142]. This will represent some very innovative audio signal processing techniques which is hoped to provide better noise reduction capability than conventional techniques.

- **Sound Synthesis Engine** - The sound synthesis engine can be substituted with FM or waveguide modeling. The rules that define the mapping between the analysis and synthesis stage have to be created from audio experts in the field of FM (with Horner research). It is planned to use the author's Yamaha VL1-m [172] as a reference waveguide model and develop pipe organ waveguide models with the audio expertise model developed in this thesis. It is believed, by the author, that digital waveguide modeling offer the most promising potential for future electronic pipe organ sound synthesis.

- **Pipe Organ Sound Position** - Recent research on pipe organ have included the sound position and studied for example aspects of loudness of pipe organ sounds at different locations in an auditorium [63] and more recently dependency of the tone timbre sensation in the standard position of the listener [152]. These 3-dimension aspects of pipe organ perception can further improved from detailed knowledge about the pipe organ sounds.

- **Advanced Sound Database** - The development of advanced sound database is needed to centralise sound data in a multichannel format and with the highest

sound quality. This can be done at the University of Plymouth together with the Opera system if perceptual evaluation of audio quality is required. Sound collection guidelines should be developed with the audio expert to formalise their format. For example more information about the equipment used for the recordings have to included, the recordings context, what pipe, contact/email of pipe organ owner or person responsible, etc. All these information will help improving the sound analysis.

7.5.3 Multi-disciplinary Research

This research has been focused on the fuzzy modeling and working with the audio experts revealed that this research work would greatly benefit from multi-disciplinary research collaborations, with contributions as follows:

- **Collaboration with researchers in Psychology** - Future research should seek advice from researchers of the Department of Psychology involved research dealing with sounds [65] [32]. It will help to understand the psychology issues about the cognitive aspects behind the judgement process of sound quality and audio equipments [113].

- **Collaboration with Organ Builders** - Collaboration with organ builders like Musicom audio experts, would greatly help future research especially with issues related to the "voicing of pipes", which is the most interesting potential use of the fuzzy model of audio expertise.

- **Collaboration with Doctors** - Collaboration with medical professionals would help understand the complexity of the human auditory system from a medical point of view and especially issues related to expert listeners (the so-called "golden ears").

- **Collaboration with Musicians** - Ultimately the Musicom ADE has been design

the use used in performance and thus the end-users, the organists and musicians should be consulted about their opinions with issues such as sound synthesis control, and audio quality.

7.5.4 Future Electronic Pipe Organs

This work will help to improve future development of electronic pipe organs both in term of software and hardware, and can consist of the integration of this research ad its results in further development of a *pipe organ modeler* and development of future hardware to update Musicom's ADE, with some *pipe organ synthesis platform*.

- **Pipe Organ Modeler** - Our industrial collaborator Musicom Ltd and the two audio experts have shown interest to integrate the results of the research into the tools they supply to their customer. Modern electronic pipe organ environment consist of the hardware sound generator like the Musicom ADE (shown in Figure 3.18) with a software that can edit and control the parameters. Client of Musicom are used to manipulate the real pipes and future work would involve more hands-on with acoustics experiments and also to try to investigate the mapping between the real change on real pipe (i.e. work of *voicing*) with the sound synthesis model. Books like [71] can provide a vast amount of real size specifications for chosen pipe in English church and cathedrals.

- **Waveguide Pipe Modeling** - Digital waveguide modeling will be investigated for the development of complex pipe structure from pipe organ as they are considered to provide the best sound quality of all sound synthesis techniques. AI-based techniques will be used to estimate and optimise the synthesis model parameters. Real-time implemented in assembler onto DSP chips will be carried out. Work described in [179] will form the basic model enhanced by the work of Välimäki [161].

- **Pipe modeling using FEM -** Novel techniques based on Finite Element Modeling (FEM) will be investigated using tool such as FEMLAB† to model air flow in the acoustic pipe. Real measurements or pipes will be carried out at organ builders collaborating with Mucicom Ltd. Other possible tool also includes ANSYS‡ could be also used for air flow simulation.

- **Pipe Organ Synthesis Platform -** Since the original design of the Musicom ADE, Digital Signal Processing technologies have progressed very much over the last 20 years and up-to-date DSPs for new pipe organ electronic musical instruments. It would be possible to design of an intelligent combination of hardware and software. The software tools that control the hardware (DSPs) do not have to be tailored like in the present case. The hardware is so powerful that there is a need to look at how the software can make full use of its flexibility. Lane and his colleagues at Motorola already demonstrated the use of DSP-based modeling of Analog model [107]. Such multiple DSP system for sound synthesis already exist, one example is the Yamaha VL1-m. It features three Yamaha proprietary DSPs chips implementing complex models of musical instruments [170]. The Kyma• and recently Capybara• system are also good examples of multiple Motorola DSPs system for sound synthesis [143]. For our industrial collaborator the idea of designing a multiple DSP system with one DSP per pipe model is not far from reality, looking at the cheap price of current DSP chips compared to the high cost of ASIC development. A DSP model of a pipe could be implemented and controlled in real-time by *intelligent* software in a similar manner to the system developed in this thesis.

†FEMLAB is a registered Trademark of COMSOL Inc.
‡ANSYS is a registered Trademark from ANSYS Inc.
•Kyma and Capybara are registered trademarks of Symbolic Sound Corp.

7.5.5 New Sound Quality Index

Future work leading to a *New Sound Quality Index* are the following:

- **Audio Expert Perceptual Model** - It would be interested to compare the audio experts hearing test with results published by Shlien and Soulodre in [146] and [147] on the auditory models of gifted listeners (the famous "Golden Ears") and on the measurement of the hearing characteristics of these experts listeners. It has been shown that audio experts have a very accurate hearing system. This is due to their extensive training in listening and assessing sound quality.

- **Analysis of PEAQ Cognitive Model** - The PEAQ cognitive model is based on a neural network used to map the Model Output Values (MOVs) into one single Objective Difference Grade (ODG). The study of neural network weights and generation of mapping surfaces (ODG versus two other MOVs) could reveal interesting aspects of the PEAQ cognitive mode. MATLAB Neural Network Toolbox will allow rapid implement the cognitive model using weights values available from the ITU-R BS1387 documents. ISO surfaces corresponding to perceptible threshold (ODG values of -1) can be generated and help understand the influence of each MOV onto the final ODG. In Chapter 2, Table 2.2 presented the 5 levels scale of the Objective Difference Grade, it must be said that levels of ODG such as *Perceptible*, *Slightly annoying* and *Annoying*, are only of interest for assessing low-cost audio quality devices in cases where the sound synthesis resources are very limited, for some multimedia and mobil phone for example. From the audio experts point of view, only the perceptible threshold is of real interest.

- **Fuzzy-based PEAQ Cognitive Model** - By studying the equivalence between neural networks and fuzzy systems as described in [46] one could investigate an equivalent fuzzy model of the PEAQ cognitive model. Of most importance and

interest is the rules by which this cognitive model emulate the way human make judgment about perceived sound quality from the Model Output Variables (MOVs) defined on the basis of audio features gained from the sound analysis by the perceptual model. One could imagine the same process applied to the Neural network of PEAQ in order to discover what real knowledge it has been modeling. It is hoped that such analysis would reveal already known rules (rationale from facts from the perceptual audio coding literature), and maybe other knowledge about audio coding that been learned by the neural network during its training.

- **Re-training of PEAQ Cognitive Model** - PEAQ neural network should be re-trained based on a sound database with sound synthesis artefacts scored by the audio experts. This will provide a more accurate prediction of sound synthesis quality, as sound synthesis artefacts are very different from those described in audio coding [38]. MATLAB Neural Network Toolbox can be used to implement and train a neural network using the scores from listening test emailed by the audio experts using the SDG feature of WinPVoc (See Figure 3.17). This would provide an accurate *audio expert evaluation of sound quality.*

- **No-Reference Sound Quality Prediction** - No-reference sound quality prediction can be developed to remove the need for both reference and test signals required in current objective method such as PEAQ. Novel techniques should be developed to predict the sound quality directly from the synthesis parameters. Using a sound synthesis technique and its known parameters, PEAQ algorithm can be used to assess the perceived quality of the synthetic sound. The sound synthesis parameters can be used to train a neural network model to learn the mapping between the sound synthesis parameters to the objective perceived sound quality.

- **Novel Sound Quality Index** - Novel sound quality index will be developed inspired from the work on perceptual quality index called Structural Similarity (SSIM)

that was proposed in [165]. The SSIM will most probably be followed by a non-linear mapping function (using MATLAB Curve Fitting Toolbox) and will attempt to approximate PEAQ ODG results within the 0.02 % tolerance as defined in the ITU-R standard [81]. This represent the most interesting aspect of the future work on sound quality.

7.5.6 Other Applications in Audio

The novel approach presented in this thesis can be applied to other of musical instruments and of interest are especially piano, bells and Hammond organ. Furthermore, the same concept can be applied to another topic of interest in audio research such as artificial reverberation.

- **Piano Audio Expertise** - Collaboration with Piano manufacturers and their audio experts will help to develop fuzzy model of audio expertise specialised for Piano. This is of interest for Musicom Ltd as the Musicom ADE can model any type of musical instrument.

- **Automated Piano Modeling** - Recent work on automated modeling of piano sounds based on Teager Energy Operator sound analysis [85] combined with multiple excitation-filter synthesis [108] and using PEAQ algorithm as a quality metric has been reported in [177]. Evaluation of piano tones sound quality as described in [160] can be used as a starting point for further investigations like in [159] and more recently in [41]. High quality data are also available from commercial CDROM or VST plug-in such as "The Grand" (Steinberg GmbH).

- **Modeling of Church Bells** - Investigation with sound analysis of church bells from Musicom industrial collaborators will provide another use to the fuzzy model for modeling church bells. Frequency Modulation (FM) described in [110] as well as digital waveguide model recently developed in [86] will constitute the basis for

this line of investigation. This should help other researchers working in the design of bells [119].

- **Perceptual Modeling of Hammond organs** - Recently donated by Musicom Ltd, an original Hammond organ [158] will be used to investigate automated perceptual modeling of musical instruments and further evaluate the intelligent system. Musicom ADE can also be used to synthesise this famous musical instrument which uses mechanical, magnetic and electronic to produce a very characteristic sound.

- **Artificial Reverberation** - Other field of audio research such as artificial reverberation could benefit from using the approach presented in this thesis. There is a vast knowledge available in the literature with audio experts such as Blesser [13] and Griesinger [52], both considered as the two most imminent experts in this field. Artificial reverberation would be of interest for both industrial collaborators. For Musicom Ltd it would allow to simulate complex acoustic environment where pipe organ are situated and for Allen & Heath it would be an important feature in future developments of digital mixing desks. Models to be used will include Feedback Delay Networks (FDN) [137] and recent development based on perceptual reduction [111].

7.6 Conclusions

An important contribution of this work has been to provide a research platform for investigating, and demonstrating the feasibility and validity of the novel Artificial Intelligence-based approach to modeling musical instruments and sound design. Most of the efforts were focused on understanding the real problems audio experts are faced with in sound analysis, sound modeling, sound synthesis and sound quality assessment, and to develop efficient solutions to these problems. The novelty in this work is in the fuzzy model of

audio expertise for sound synthesis parameter optimisation and in the automated intelligent audio system that provide an objective prediction of sound synthesis quality. This research has allowed, for the first time in computer music, the capture and exploitation of audio knowledge to help improve the complex and time-consuming process of modeling musical instruments and designing sounds. To the author's knowledge, this approach is unique and has no equivalent in audio research. The fuzzy model of audio expertise developed in the project has a basic set of rules that will require extensive validation by audio experts before it is made available as a software tool.

References

[1] Aarts, E. and Korst, J. (1996). *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing.* Cambridge, MA: MIT Press. ISBN 0-471-92146-7.

[2] Aarts, R. M. and Dekkers, R. T. (1999). A Real-Time Speech-Music Discriminator. *Journal of the Audio Engineering Society*, 47(9):720–725.

[3] Akai Professional Musical Instruments Corporation website (2004). `http://www.akaipro.com`.

[4] Allen-Heath Ltd website (2003). `http://www.allen-heath.com`.

[5] Andersen, T. H. and Jensen, K. (2001). On the Importance of Phase Information in Additive Analysis/Synthesis of Binaural Sounds. *Proceedings of the 2001 International Computer Music Conference (ICMC2001), Havana, Cuba.*

[6] Ando, S. and Yamaguchi, K. (1993). Statistical Study of Spectral Parameters in Musical Instruments Tones. *Journal of the Acoustic Society of America*, 94(1):37–45.

[7] Atmel Corp. (2002). *ATSAM9707 - Integrated sound processor studio*, Atmel Corp. edition.

[8] Atmel Corp. website (2003). `http://www.atmel.com`.

[9] Ayers, L. and Horner, A. B. (1999). Modeling the Woodstock and Gamelan for Synthesis. *Journal of the Audio Engineering Society*, 47(10):813–823.

[10] Baron, S. D. and Gil, D. A. (1999). The CPLD as a General Physical Modeling Synthesis Engine. *Proceedings of the 2nd COST-G6 Workshop on Digital Audio Effects (DAFX99), Trondheim, Norway, December 9-11, 1999*, pages 121–124.

[11] Benjamin, E. (2002). Evaluating Digital Audio Artifacts with PEAQ. *Proceedings of the 113th Audio Engineering Society Convention, Los Angeles, CA, USA, October 5-8, 2002.* Preprint 5711.

[12] Bertini, G., Magrini, M., and Tarabella, L. (2000). Spectral Data Management Tools for Additive Synthesis. *Proceedings of the Conference on Digital Audio Effects (DAFX00), Verona, Italy, December 7-9, 2000.*

[13] Blesser, B. (2001). An Interdisciplinary Integration of Reverberation. *Proceedings of the 111th Audio Engineering Society Convention, New York, USA, Nov. 30 - Dec. 3, 2001.* Preprint 5468.

[14] Bosi, M. (1999). Filter Banks in Perceptual Audio. *Proceedings of the AES 17-th International Conference on High Quality Audio Coding, Florence, September 1999*, pages 125–135. Tutorial 2.

[15] Brandenburg, K. (1999). MP3 and AAC Explained. *Proceedings of the AES 17-th International Conference on High Quality Audio Coding, Florence, September 1999*, pages 99–110.

[16] Brandenburg, K. and Bosi, M. (1997). Overview of mpeg audio: Current and future standards for low bit-rate audio coding. *Journal of the Audio Engineering Society*, 45(1/2):4–19.

[17] Brandenburg, K. and Stoll, G. (1994). ISO/MPEG-1 Audio: A Generic Standard for Coding of High-Quality Digital Audio. *Journal of the Audio Engineering Society*, 42(10):780–792.

[18] Bristow-Johnson, R. (2003). Letters to the editor: Comments on A Simplified Wavetable Matching Method Using Combinatorial Basis Spectra Selection. *Journal of the Audio Engineering Society*, 51(3):162–163.

[19] Chowning, J. M. (1977). Method of Synthesizing a Musical Sound. U.S. Patent 4018121.

[20] Coleman, T., Branch, M. A., and Grace, A. (1999). *MATLAB Optimization Toolbox - User's Guide Version 2.*

[21] Comerford, P. J. (1980). Digital Generator for Musical Notes. U.S. Patent 4202234.

[22] Comerford, P. J. (1981). Bradford Musical Instrument Simulator. *IEE Proceedings*, 128(5):364–372. Pt. A.

[23] Comerford, P. J. (1987). Further Developments of the Bradford Musical Instrument Simulator. *IEE Proceedings*, 134(10):799–806. Pt. A.

[24] Comerford, P. J. (1993). Simulating an Organ with Additive Synthesis. *Computer Music Journal*, 17(2):55–65.

[25] Cox, E. (1994). *The Fuzzy Systems Handbook A practitioner's guide to building, using, and maintaining fuzzy systems.* Academic Press Bardon Entreprises Portsmouth. ISBN 0-121-94270-8.

[26] De Bernadinis, F., Roncella, R., Saletti, R., Terremi, P., and Bertini, G. (1997). A Single-Chip 1200 Sinusoid Real-Time Generator for Additive Synthesis of Musical Signals. *Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, Munich, Germany*, 1:427–430.

[27] Desainte-Catherine, M. and Marchand, S. (2000). High Precision Fourier Analysis of Sounds Using Signal Derivatives. *Journal of the Audio Engineering Society*, 48(7/8):654–667.

[28] Ding, Y. and Qian, X. (1997). Processing of Musical Tones Using a Combined Quadratic Polynomial-Phase Sinusoid and Residual (QUASAR) Signal Model. *Journal of the Audio Engineering Society*, 45(7/8):571–584.

[29] Disley, A. C. and Howard, D. M. (2003). Timbral Semantics and the Pipe Organ. *Proceedings of the Stockholm Music Acoustic Conference (SMAC2003), Stockholm, Sweden, August 6-9, 2003*, pages 607–610.

[30] E-MU Digital Audio Systems website (2004). `http://www.emu.com`.

[31] Eberlein, E., Gerhauser, H., Popp, H., Seitzer, D., Schott, H., and Brandenburg, K. H. (1998). Digital Adaptive Transformation Coding Method. U.S. Patent 5742735.

[32] Edworthy, J. and Meredith, C. (1994). Cognitive psychology and the design of alarm sounds. *Medical Engineering & Physics*, 16:445–449.

[33] Egusa, Y., Akahori, H., Morimura, A., and Wakami, N. (1995). An application of fuzzy set theory for an electronic video camera image stabilizer. *IEEE Transactions on Fuzzy Systems*, 3(3):351–356.

[34] Emu Systems Inc. (1993). *Morpheus - Operation Manual*, Emu Systems Inc. edition. F1420 Rev. C - Manual - Riley Smith.

[35] Encyclopedia of Organ Stops (2003). `http://www.organstops.org`.

[36] EPSRC Grant GR/S00859/01 (2003). Intelligent and Perceptual-based Techniques for Automated Design and Synthesis of Musical Instrument Sounds. `http://www.tech.plymouth.ac.uk/spmc/S00859/`.

[37] Erne, M. (1998). Digital Audio Compression Algorithms. *Proceedings of the 1st COST-G6 Workshop on Digital Audio Effects (DAFX98), Barcelona, Spain*, pages 99–110.

[38] Erne, M. (2001). Perceptual Audio Coders What to listen for. *Proceedings of the 111th Audio Engineering Society Convention, New York, USA, Nov. 30 - Dec. 3, 2001.* Preprint 5489.

[39] Fletcher, N. H. and Rossing, T. D. (1998). *The Physics of Musical Instruments.* Springer-Verlag, second edition. ISBN 0-38798-374-0.

[40] Frigo, M. and Johnson, S. G. (1998). FFTW: An Adaptive Software Architecture for the FFT. *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech, and Signal Processing, Seattle, Washington, USA*, 3:1381–1384.

[41] Galembo, A. and Askenfelt, A. (2003). Phase Randomisation in Piano Bass Tones. *Proceedings of the Stockholm Music Acoustic Conference (SMAC2003), Stockholm, Sweden, August 6-9, 2003*, pages 151–154.

[42] Garibaldi, J. M. (1997). *Intelligent Techniques for Handling Uncertainty in the Assessment of Neonatal Outcome.* Ph.D. thesis, Department of Electrical, Communications and Electronic Engineering (DECEE), University of Plymouth, Drake Circus, Plymouth PL4 8AA, Devon, United Kingdom.

[43] Garibaldi, J. M. and Ifeachor, E. C. (1999). Application of Simulated Annealing Fuzzy Model Tuning to Umbilical Cord Acid-Base Interpretation. *IEEE Transactions on Fuzzy Systems*, 7(1):72–84.

[44] Garibaldi, J. M., Ifeachor, E. C., Szczepaniak, P. S., Lisboa, P. J., and Kacprzyk, J. (2000). The development of a fuzzy expert system for the analysis of umbilical cord blood. *Fuzzy Systems in Medicine*, pages 652–668. Springer-Verlag.

[45] Garibaldi, J. M., Westgate, J. A., and Ifeachor, E. C. (1999). The Evaluation of an Expert System for the Analysis of Umbilical Cord Blood. *Artificial Intelligence in Medicine*, 17(2):109–130.

[46] Gaweda, A. E. and Zurada, J. M. (2001). Equivalence Between Neural Networks and Fuzzy Systems. *Proceedings of the International Joint Conference for Neural Networks (IJCNN2001), Washington, USA, July 15-19, 2001*, pages 1334–1339.

[47] George, E. B. and Smith, M. J. T. (1997). Speech Analysis/Synthesis and Modification using an Analysis-by-Ssynthesis/Overlap-Add Sinusoidal Model. *IEEE Transactions on Speech and Audio Processing*, 5(5):389–406.

[48] Godsill, S. J. and Rayner, P. J. W. (1998). *Digital Audio Restoration - a Statistical Model-based Approach*. Springer-Verlag London Limited. ISBN 3-540-76222-1.

[49] Gokhale, M., Frigo, J., and Lavenier, D. (2003). Experience with a Hybrid Processor: K-Means Clustering. *The Journal of Supercomputing*, 26(2):131–148.

[50] Goossens, M., Mittelbach, F., and Samarin, A. (1993). *The LaTeX Companion*. Addison-Wesley Pub Co., second edition. ISBN 0-201-54199-8.

[51] Grey, J. M. (1977). Multidimensional Perceptual Scaling of Musical Timbres. *Journal of the Acoustic Society of America*, 61(5):1270–1277.

[52] Griesinger, D. (2002). Stereo and Surround Panning in Practice. *Proceedings of the 112th Audio Engineering Society Convention, Munich, Germany, May 10-13, 2002*. Preprint 5564.

[53] Grill, B., Brandenburg, K. H., Sporer, T., Kurten, B., and Eberlein, E. (1996). Digital Encoding Process. U.S. Patent 5579430.

[54] Grinstein, G., Trutschl, M., and Cvek, U. (2001). High-Dimensional Visualizations. *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining (KDD-2001), San Francisco, California, August 26-29, 2001*, pages 1–14.

[55] Grizard, J. (2003). *Sound Synthesis using Independent Component Analysis.* M. Sc. Thesis, Department of Communications and Electronic Engineering (DCEE), University of Plymouth, Drake Circus, Plymouth PL4 8AA, Devon, United Kingdom.

[56] Hamadicharef, B. A. (2003a). Clustering Algorithms for Group Additive Synthesis. Internal Report 4, School of Computing, Communications and Electronic (SoCCE), University of Plymouth, Drake Circus, Plymouth PL4 8AA, Devon, United Kingdom.

[57] Hamadicharef, B. A. (2003b). Sound Analysis of Pipe Organ Database. Internal Report 3, School of Computing, Communications and Electronic (SoCCE), University of Plymouth, Drake Circus, Plymouth PL4 8AA, Devon, United Kingdom.

[58] Hamadicharef, B. A. (2003c). WinPVoc - User's Guide. Internal Report 2, School of Computing, Communications and Electronic (SoCCE), University of Plymouth, Drake Circus, Plymouth PL4 8AA, Devon, United Kingdom.

[59] Hamadicharef, B. A. and Ifeachor, E. C. (1999). Artificial Intelligence based Modeling of Musical Instruments. *Proceedings of the 2nd COST-G6 Workshop on Digital Audio Effects (DAFX99), Trondheim, Norway, December 9-11, 1999*, pages 117–120.

[60] Hamadicharef, B. A. and Ifeachor, E. C. (2001). An Intelligent System Approach to Sound Synthesis Parameter Optimisation. *Proceedings of the 111th Audio Engineering Society Convention, New York, USA, Nov. 30 - Dec. 3, 2001.* Preprint 5484.

[61] Hamadicharef, B. A. and Ifeachor, E. C. (2003). Objective Prediction of Sound Synthesis Quality. *Proceedings of the 115th Audio Engineering Society Convention, New York, USA, October 10-13, 2003.* Preprint 5958.

[62] Hanson, H., Maragos, P., and Potamianos, A. (1994). A system for finding speech formants and modulations via energy separation. *IEEE Transactions on Speech and Audio Processing*, 2(3):436–443.

[63] Harrison, J. M. and Thompson-Allen, N. (2000). Loudness of Pipe Organ Sounds at Different Location in an Auditorium. *Journal of the Acoustic Society of America*, 108(1):389–399.

[64] Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation.* Macmillan Publishing Company. ISBN 0-023-52761-7.

[65] Hellier, E. J., Edworthy, J., and Dennis, I. (1993). Improving auditory warning design: Quantifying the effects of different warning parameters on perceived urgency. *Human Factors*, 33(4):693–706.

[66] Hodes, T. and Freed, A. (1999). Second-order Recursive Oscillators for Musical Additive Synthesis Applications on SIMD and VLIW Processors. *Proceedings of the 1999 International Computer Music Conference (ICMC1999), Beijing, China*, pages 74–77.

[67] Hodes, T., Hauser, J., Freed, A., Wawrzynek, J., and Wessel, D. (1999). A Fixed-point Recursive Digital Oscillator for Additive Synthesis of Audio. *Proceedings of the 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing, Phoenix, Arizona, USA*, pages 993–996.

[68] Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems.* Cambridge, MA: MIT Press. ISBN 0-262-58111-6.

[69] Honda, T. (1998). Electric Violin. U.S. Patent D-403-013.

[70] Hongwei, W. and Mendel, J. M. (2002). Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems*, 10(5):622–639.

[71] Hopkins, E. J. and Rimbault, E. F. (2000). *The Organ - Its History and Construction.* Bardon Entreprises Portsmouth. ISBN 0-952-81846-9.

[72] Horner, A. B. (1998). Nested Modulator and Feedback FM Matching of Instrument Tones. *IEEE Transactions on Speech and Audio Processing*, 6(4):398–409.

[73] Horner, A. B. (2001). A Simplified Wavetable Matching Method Using Combinatorial Basis Spectra Selection. *Journal of the Audio Engineering Society*, 46(11):1060–1066.

[74] Horner, A. B., Ayers, L., and Law, D. (1996). Modeling Small Chinese and Tibetan Bells. *Journal of the Audio Engineering Society*, 45(3):148–159.

[75] Horner, A. B., Ayers, L., and Law, D. (1999). Synthesis Modeling of the Chinese Dizi, Bawu and Sheng. *Journal of the Audio Engineering Society*, 47(12):1076–1087.

[76] Hyvärinen, A. and Oja, E. (2000). Independent Component Analysis: Algorithms and Applications. *Neural Networks*, 13(4-5):411–430.

[77] Ifeachor, E. C., Curnow, J. S. K., Outram, N. J., and Skinner, J. F. (2001). Models for Handling Uncertainty in Fetal Heart Rate and ECG Analysis. *Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS'01), Istanbul, Turkey, October 25-28, 2001.*

[78] Ifeachor, E. C. and Jervis, B. W. (2002). *Digital Signal Processing: A Practical Approach.* Prentice Hall, second edition. ISBN 0-201-59619-9.

[79] ISO/IEC JTC 1/SC 29/WG 11 N2503-sec5 ((1999)). MPEG-4 Final Draft on Structured Audio.

[80] ITU-R Recommendation BS.1116 (1997). Methods for the subjective assessment of small impairments in audio systems including multi-channel sound systems.

[81] ITU-R Recommendation BS.1387 ((1998)). Method for Objective Measurements of Perceived Audio Quality (PEAQ).

[82] ITU-R Recommendation BS.562-3 (1997). Subjective Assessment of Sound Quality.

[83] ITU-R Recommendation P.862 (2000). Perceptual Evaluation of Speech Quality (PESQ), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs.

[84] Jensen, K. (1999). Envelope Model for Isolated Musical Sounds. *Proceedings of the 2nd COST-G6 Workshop on Digital Audio Effects (DAFX99), Trondheim, Norway, December 9-11, 1999*, pages 35–39.

[85] Kahrs, M. (2001). Audio Applications of the Teager Energy Operator. *Proceedings of the 111th Audio Engineering Society Convention, New York, USA, Nov. 30 - Dec. 3, 2001*. Preprint 5473.

[86] Karjalainen, M., Välimäki, V., and Esquef, P. A. A. (2002). Efficient Modeling and Synthesis of Bell-like Sounds. *Proceedings of the 2002 DAFX Conference, Hamburg, Germany, September 26-28, 2002*, pages 181–186.

[87] Karnik, N. N., Mendel, J. M., and Qilian, L. (1999). Type-2 Fuzzy Logic Systems. *IEEE Transactions on Fuzzy Systems*, 7(6):643–658.

[88] Karplus, K. J. and Strong, A. R. (1983). Digital Synthesis of Plucked-String and Drum Timbres. *Computer Music Journal*, 7(2):42–55.

[89] Keiler, F. and Marchand, S. (2002). Survey on Extraction of Sinusoids in Stationary Sounds. *Proceedings of the 2002 DAFX Conference, Hamburg, Germany, September 26-28, 2002*, pages 51–58.

[90] Keith, R. D. F., Greene, K. R., Ifeachor, E. C., and Westgate, J. (1997). Foetal Monitor. U.S. Patent 5609156.

[91] Keyhl, M., Schmidmer, C., and Wachter, H. (1999). A combined measurement tool for the objective perceptual based evaluation of the compressed speech and audio signals. *Proceedings of the 106th Audio Engineering Society Convention, Munich, Germany, May 8-11, 1999.* Preprint 4931.

[92] Kleczkowski, P. (1989). Group Additive Synthesis. *Computer Music Journal*, 13(1):12–20.

[93] Kniest, J. and Petersen, J. D. (1998). Digital Tone Synthesis Modeling for Complex Instruments. U.S. Patent 5747714.

[94] Kokkelmans, S., Verge, M. P., Hirschberg, A., Wijnands, A. P., and Schoffelen, R. (1999). Acoustic Behavior of Chimney Pipes. *Journal of the Acoustic Society of America*, 105(1):546–551.

[95] Komano, T. and Kunimoto, T. (1994). Electronic musical instrument which simulates physical interaction of piano string and hammer. U.S. Patent 5182415.

[96] Koorlander, T. (1998). *Voicing Assembler User Manual*, Musicom Ltd edition.

[97] Koorlander, T. (1998-2004). Personal correspondence.

[98] Koorlander, T. (2002). Organ Vitals Get Electric Treatment. *EE Times*. Article.

[99] Koorlander, T. (2003). *VASM - User Manual*, Musicom Ltd edition. (Voicing Assembler Hyperlink).

[100] Koorlander, T. and Blyth, G. (2000). *Musicom ADE System - Brochure*, Musicom Ltd edition.

[101] Kostek, B. (1995a). Feature Extraction Methods for the Intelligent Processing of Musical Instruments. *Proceedings of the 99th Audio Engineering Convention, New York, USA, October 6-9, 1995.* Preprint 4076.

[102] Kostek, B. (1995b). Statistical versus Artificial Intelligence based Processing of Subjective Test Results. *Proceedings of the 98th Audio Engineering Convention, Paris, France, February 25-28, 1995.* Preprint 4018.

[103] Kunimoto, T. (1993a). Brass Instrument Type Tone Synthesizer. U.S. Patent 5272275.

[104] Kunimoto, T. (1993b). Musical Tone Synthesizing Device. U.S. Patent 5182415.

[105] Kunimoto, T. (1998). Engine Exhaust Sound Synthesizer. U.S. Patent 5835605.

[106] Lagrange, M. and Marchand, S. (2001). Real-Time Additive Synthesis of Sounds by Taking Advantage of Psychoacoustics. *Proceedings of the Conference on Digital Audio Effects (DAFX01), Limerick, Ireland, December 6-8, 2001.*

[107] Lane, J., Hoory, D., Martinez, E., and Wang, P. (1997). Modeling Analog Synthesis with DSPs. *Computer Music Journal*, 21(4):32–41.

[108] Laroche, J. and Meillier, J.-L. (1994). Multichannel Excitation/Filter Modeling of Percussive Sounds with Application to the Piano. *IEEE Transactions on Speech and Audio Processing*, 2(2):329–344.

[109] Laurenti, N. and De Poli, G. (2000). A Method for Spectrum Separation and Envelope Estimation of the Residual in Spectrum Modeling of Musical Sound. *Proceedings of the Conference on Digital Audio Effects (DAFX00), Verona, Italy, December 7-9, 2000*, pages 233–236.

[110] Lee, K. and Horner, A. B. (1999). Modeling Piano Tones with Group Synthesis. *Journal of the Audio Engineering Society*, 47(3):101–111.

[111] Lee, W.-C., Liu, C.-M., Yang, C.-H., and Guo, J.-I. (2003). Fast Perceptual Convolution for Room Reverberation. *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx-03), London, UK, September 8-11, 2003.*

[112] Lim, S. M. and Tan, B. T. G. (1999). Performance of the Genetic Annealing Algorithm in DFM Synthesis of Dynamic Musical Sound Samples. *Journal of the Audio Engineering Society*, 47(5):339–354.

[113] Martens, W. L. and Atsushi, M. (2002). Multidimensional Perceptual Scaling of Tone Color Variation in Three Modeled Guitar Amplifiers. *Proceedings of the 112th Audio Engineering Society Convention, Munich, Germany, May 10-13, 2002.* Preprint 5552.

[114] Martin, K. D. and Kim, Y. E. (1998). Musical Instrument Identification: A pattern-recognition approach. *Proceedings of the 136th meeting of the Acoustical Society of America, October 13, 1998.*

[115] Masuda, H. and Kunimoto, T. (1995). Wind Type Tone Synthesizer Adapted for Simulating a Conical Resonance Tube. U.S. Patent 5438156.

[116] Mattila, V.-V. (2001). Descriptive analysis of speech quality in mobile communications: Descriptive language development and external preference mapping. *Proceedings of the 111th Audio Engineering Society Convention, New York, USA, Nov. 30 - Dec. 3, 2001.* Preprint 5455.

[117] McAdams, S., Beauchamp, J. W., and Meneguzzi, S. (1999). Discrimination of Musical Instrument Sounds Resynthesized with Simplified Spectrotemporal Parameters. *Journal of the Acoustic Society of America*, 105(2):882–897.

[118] McAulay, R. J. and Quatieri, T. F. (1986). Speech Analysis/Synthesis Based on a Sinusoidal Representation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(4):744–754.

[119] McLachlan, N. and Cabrera, D. (2002). Calculated Pitch Sensations for new Musical Bell Designs. *Proceedings of the 7th International Conference on Music Perception and Cognition (ICMPC7), Sydney, Australia, July 17-21, 2002*, pages 600–603.

[120] Mendel, J. M. and John, R. I. B. (2002). Type-2 fuzzy sets made simple. *IEEE Transactions on Fuzzy Systems*, 10(2):117–127.

[121] Menzies, D. and Howard, D. (1988). The CyberWhistle: An Instrument For Live Performance. *In Proceedings of the XII Colloquium for Musical Informatics. Gorizia, Italy.*

[122] Musicom PipeSpecBuilder 1.6 (2000). *Specification Manager version 2.65*, Musicom Ltd edition.

[123] Oates, S. and Eaglestone, B. M. (1997). Analytical Methods for Group Additive Synthesis. *Computer Music Journal*, 21(2):21–40.

[124] Okamura, S. (2000). Electric Cello. U.S. Patent D-419587.

[125] Ozen, T. and Garibaldi, J. M. (2003). Investigating Adaptation in Type-2 Fuzzy Logic Systems Applied to Umbilical Acid-Base Assessment. *Proceedings of the 2003 European Symposium on Intelligent Technologies (EUNITE 2003), Oulu, Finland*, pages 289–294.

[126] Painter, T. and Spanias, A. (2000). Perceptual Coding of Digital Audio. *Proceedings of IEEE*, 88(4):451–513.

[127] PIPORG-L - Electronic mailing list (Pipe Organs and related tipcs) (2003). `http://www.albany.edu/piporg-l/`.

[128] Pollard and Jansson (1982). A Tristimulus Method for the Specification of Musical Timbre. *Acustica*, 51:162–171.

[129] Portnoff, M. R. (1976). Implementation of the digital phase vocoder using the fast Fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(3):243–248.

[130] Portnoff, M. R. (1980). Time-frequency representation of digital signals and systems based on short-time fourier analysis. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(1):55–69.

[131] Puckette, M. S. and Brown, J. C. (1998). Accuracy of Frequency Estimates Using the Phase Vocoder. *IEEE Transactions on Speech and Audio Processing*, 6(2):166–176.

[132] Rabiner, L., Cheng, M., Rosenberg, A., and McGonegal, C. (1976). A Comparative Performance Study of Several Pitch Detection Algorithms. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(5):399 –418.

[133] Rioux, V. (2001). *Sound Quality of Flue Organ Pipes - An Interdisciplinary Study on the Art of Voicing.* Ph.D. Thesis, School of Civil Engineering, Department of Applied Acoustics, Chalmers University of Technology, SE-41296 Gothenburg, Sweden. Report F 01-02.

[134] Rioux, V. and Västfjäll, D. (2001). Analyses of Verbal Descriptions of the Sound of the Flue Organ Pipe. *MusicæScientiæ*, 5(1).

[135] Röbel, A. (2003a). A New Approach to Transient Processing in the Phase Vocoder. *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx-03), London, UK, September 8-11, 2003*, pages 344–349.

[136] Röbel, A. (2003b). Transient Detection and Preservation in the Phase Vocoder. *Proceedings of the 2003 International Computer Music Conference (ICMC2003), Singapore, Sept. 29 - Oct. 4, 2003*, pages 247–250.

[137] Rocchesso, D. (1997). Maximally Diffusive Yet Efficient Deedback Delay Networks for Artificial Reverberation. *IEEE Signal Processing Letters*, 4(9):252–255.

[138] Rodgers Instruments website - (now part of the Roland Group of musical instrument companies) (2003). `http://www.rodgersinstruments.com`.

[139] Roger Jang, J.-S. and Gulley, N. (1995). *MATLAB Fuzzy Logic Toolbox - User's Guide Version 2.*

[140] Rossum, D. P. (2002). Digital Sampling Instrument Employing Cache Memory. U.S. Patent 6365816.

[141] Russo, F. (1992a). A Fuzzy Approach to Digital Signal Processing: Concepts and Applications. *IEEE Transactions on Instrumentation and Measurement*, 45(2):640–645.

[142] Russo, F. (1992b). A User-Friendly Research Tool For Image Processing With Fuzzy Rules. *Proceedings of the 1992 IEEE International Conference on Fuzzy Systems (Fuzz-IEEE 1992), San Diego, USA*, pages 561–568.

[143] Scaletti, C. (1989). The Kyma/Platypus Computer Music Workstation. *Computer Music Journal*, 13(2):23–38.

[144] Schantz Organ Company website (2005). `http://www.schantzorgan.com`.

[145] Serra, X. and Smith, J. O. (1991). Musical Synthesizer Combining Deterministic and Stochastic Waveforms. U.S. Patent 5029509.

[146] Shlien, S. (2000). Auditory Models for Gifted Listeners. *Journal of the Audio Engineering Society*, 50(1-2):1032–1044.

[147] Shlien, S. and Soulodre, G. (1996). Measuring the Characteristics of Experts Listeners. *Proceedings of the 101st Audio Engineering Society Convention, Los Angeles, CA, USA, November 8-11, 1996.* Preprint 4339.

[148] Smith, J. O. (1991). Digital Signal Processing Using Waveguide Networks. U.S. Patent 4984276.

[149] Smith, J. O. (1992). Physical Modeling Using Digital Waveguides. *Computer Music Journal*, 16(4):74–87.

[150] Smith, J. O. and Van Duyne, S. A. (1995). Commuted Piano Synthesis. *Proceedings of the 1995 International Computer Music Conference (ICMC1995), Banff, Canada, September 3-7, 1995*, pages 335–342.

[151] Strong, A. R. and Karplus, K. J. (1987). Wavetable-Modification Instrument and Method for Generating Musical Sound. U.S. Patent 4649783.

[152] Syrovy, V., Otcenasek, Z., and Stepanek, J. (2003). Subjective Evaluation of Organ Pipe Timbre in the Standard Listener Positions. *Proceedings of the Stockholm Music Acoustic Conference (SMAC2003), Stockholm, Sweden, August 6-9, 2003*, pages 333–336.

[153] Tan, B. T. G. and Lim, S. M. (1996). Automated parameter optimization for double frequency modulation synthesis using the genetic annealing algorithm. *Journal of the Audio Engineering Society*, 44(1/2):3–15.

[154] Taniwaki, T. (1999). Woodwind-styled Electronic Musical Instrument. U.S. Patent 5922985.

[155] Thiede, T., Treurniet, W. C., Bitto, R., Schmidmer, C., Sporer, T., Beerends, J. G., Colomes, C., Keyhl, M., Stoll, G., Brandenburg, K., and Feiten, B. (2000). PEAQ -

The ITU Standard for Objective Measurement of Perceived Audio Quality. *Journal of the Audio Engineering Society*, 48(1/2):3–29.

[156] Tolonen, T., Välimäki, V., and Karjalainen, M. (1998). *Evaluation of Modern Sound Synthesis Methods*, volume Report 48. Helsinki University of Technology, Espoo, Finland, Helsinki University of Technology, Espoo, Finland. ISBN 9-512-24012-2.

[157] Tucker, W. and Bates, R. (1978). A Pitch Estimation Algorithm for Speech and Music. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(6):597–604.

[158] Vail, M. (1997). *The Hammond Organ – Beauty in the B*. Backbeat Books. ASIN 0879304596.

[159] Valenzuela, M. N. (1998). *Psychoacoustic model of calculating the sound quality of piano tones*. Ph.D. thesis, Technische Universität München, D-80290 München, Germany. (In German).

[160] Valenzuela, M. N. (1999). *Chapter in book: Perceived differences and quality judgments of piano sounds*, page 268273. VCH-Verlag Weinheim.

[161] Välimäki, V. (1995). *Discrete-Time Modeling of Acoustic Tubes Using Fractional Delay Filters*. Ph.D. thesis, Laboratory of Acoustics and Audio Signal Processing, Helsinki University of Technology, Otakaari 5 A, 02150 Espoo, Finland. Report 37.

[162] Verge, M.-P., Fabre, B., Mahu, W., Hirschberg, A., Van Hassel, R., and Wijnands, A. (1994). Jet formation and jet velocity fluctuations in a flue organ pipe. *Journal of the Acoustic Society of America*, 95(2):1119–1132.

[163] Vuori, J. and Välimäki, V. (1993). Parameter Estimation of Non-Linear Physical Models by Simulated Evolution-Application to the Flute Model. *Proceedings of the 1993 International Computer Music Conference (ICMC1993), Tokyo, Japan.*

[164] Wang, Z. and Bovik, A. C. (2002). A Universal Image Quality Index. *IEEE Signal Processing Letters*, 9(3):81–84.

[165] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. (2004). Image Quality Assessment: From Error Measurement to Structural Similarity. *IEEE Transactions on Image Processing*, 13(1).

[166] Wehn, K. (1998). Using Ideas from Natural Selection to Evolve Synthesized Sounds. *Proceedings of the 1st COST-G6 Workshop on Digital Audio Effects (DAFX98), Barcelona, Spain*, pages 159–167.

[167] Wolfe, P. J. and Godsill, S. J. (2003a). A Perceptually Balanced Loss Function for Short-Time Spectral Amplitude Estimation. *Proceedings of the 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, Hong Kong*, pages 425–428.

[168] Wolfe, P. J. and Godsill, S. J. (2003b). Audio Signal Processing using Complex Wavelets. *Proceedings of the 114th Audio Engineering Society Convention, Amsterdam, The Netherlands, March 22-25, 2003.* Preprint 5829.

[169] Wun, C.-W. and Horner, A. B. (2001). Perceptual Wavetable Matching for Synthesis of Musical Instrument Tones. *Journal of the Audio Engineering Society*, 49(4):250–262.

[170] Yamaha Corp. (1993a). *VL1-m - Service Manual (CSM861)*, Yamaha Corp. edition.

[171] Yamaha Corp. (1993b). *YMF262 FM Operator Type L3 (OPL3)*, Yamaha Corp. edition.

[172] Yamaha Corp. (1994). *VL1-m - Owner's Manual (Version 2)*, Yamaha Corp. edition.

[173] Yamaha Corp. website (2005). `http://www.yamaha.com`.

[174] Yuen, J. and Horner, A. B. (1997). Hybrid Sampling - Wavetable Synthesis with Genetic Algorithms. *Journal of the Audio Engineering Society*, 45(5):316–330.

[175] Zadeh, L. (1996). Fuzzy Logic = Counting with words. *IEEE Transactions on Fuzzy Systems*, 4(2):103–111.

[176] Zadeh, L. A. (1983). The role of fuzzy logic in the management of uncertainty in expert systems. *Fuzzy Sets and Systems*, 11:199–227.

[177] Zhang, Y. (2003). *Automated Modeling and Synthesis of Piano Sounds.* M. Res. Thesis, Department of Communications and Electronic Engineering (DCEE), University of Plymouth, Drake Circus, Plymouth PL4 8AA, Devon, United Kingdom.

[178] Zheng, H. and Beauchamp, J. (1999). Analysis and Critical-Band-Based Group Wavetable Synthesis of Piano Tones. *Proceedings of the 1999 International Computer Music Conference (ICMC1999), Beijing, China*, pages 9–12.

[179] Zielinski, S. K. (1996). Digital Waveguide Modeling vs. Mathematical Modeling of Organ Flue Pipe. *Proceedings of the 100th Audio Engineering Society Convention, Copenhagen, Denmark, May 11-14, 1996.* Preprint 4170.

[180] Zwicker, E. and Fastl, H. (1999). *Psychoacoustics - Facts and Models.* Springer Verlag - Series in Information Sciences, second edition. ISBN 3-540-65063-6.

Appendix

A.1 Sound Database

The sound database consists of a collection of pipe organ recordings (nearly 2GBytes in total) collected during visits all around the world to pipe organ manufacturers, churches and cathedrals (mostly in the United States) in which Musicom ADE systems are now installed and replacing or supporting the original acoustic pipe organ.

Organ: 1stCongregational

1stCongregational (United States) - This sound bank represents 220 MB of sound data, consists of 1 register (Misc), made of 15 stops (Viola, Vcele, Tierce, Princ2, Prin8, Nazard, Nasflut, Larigot, Krummhn, Kopfl4, Gemshorn, Gedk16, Gedeckt, Fldolce, Fldcel), with a total of 840 recordings.

Organ: Casavant

Casavant (Canada) - This sound bank represents 30 MB of sound data, consists of 4 registers (Swell, Pedal, Great, Choir), made of 38 stops (Violga8, Violce8, Trom8, Prin8, Plenmix, Octavin2, Oct4, Oboe8, Floct4, Bour16, Trom8, Sbass16, Oct8, Oct4, Montre16, Fournitu, Clar4, Bomb16, Trum8, Quint223, Pres4, Montre8, Fouriv2, Flute8, Flcon4, Doub2, Corniii, Bour16, Tier135, Qnasa2, Prin4, Nasa223, Krumm8, Flute4, Erzce8, Erz8, Cymiv, Bour8), with a total of 378 recordings.

Organ: EpiscopalHouston

EpiscopalHouston (United States) - This sound bank represents 126 MB of sound data, consists of 4 registers (Swell, Positiv, Pedal, Great), made of 40 stops (Voxhum8, Violgam8, Violcel8, Trompet8, Terz135, Quint113, Itaprin4, Hautboi8, Fifteen2, Cordunu8, Bourd16, Trirega8, Tierc135, Prestan8, Octave4, Naza223, Gedeckt8, Dulzia16,

Doublet2, Cromor8, Baarpfe4, Trompet8, Trommet8, Trombo16, Supoct4, Octave8, Clairon4, Bour32, Bour16, Trompet8, Trommet8, Supoct2, Spirefl8, Prest16, Octave8, Octave4, Gambe8, Fluhar8, Clairon4, Chimflu4), with a total of 718 recordings.

Organ: German

German (Germany) - This sound bank represents 12 MB of sound data, consists of 1 register (Misc), made of 9 stops (Voxhum8, Violflt8, Singcor2, Sifflt1, Quint84, Princip4, Prin, Flote8, Bombar32), with a total of 88 recordings.

Organ: Hexham

Hexham (England) - This sound bank represents 17 MB of sound data, consists of 1 register (Misc), made of 14 stops (Trump4, Tromp8, Schal8, Rohr8, Rohr4, Prin8, Oct4, Gamb8, Gamb4, Fif2, Dul8, Cged8, Bourdon, Basson16), with a total of 97 recordings.

Organ: Houston

Houston (United States) - This sound bank represents 270 MB of sound data, consists of 7 registers (Swell, Positiv, Pedal, Great, Fanfare, Echo, Choir), made of 67 stops (Violone, Violedeg, Violacel, Trompett, Tierce, Spillflo, Principa, Octavin, Nazard, Hautbois, Geigen, Gedeckt, Flutecel, Flautdol, Clairon, Basson, Spitzgei, Quintato, Prinzipa, Principa, Oktav, Nasat, Nachthor, Krummhor, Gemshorn, Gedepom, Dulzian, Rohrflot, Principa, Posaune, Contposa, Contgeig, Conposcl, Congeicl, Chorbass, Bourdon, Bombarde, Blockflo, Trompett, Spitzflo, Principa, Oktave, Octave, Koppelfl, Grosstie, Grossqui, Fltharm, Fagotto, Bordun, Granprin, Granocta, Granfift, Voxhum, Viola, Flutestp, Concflut, Zartflot, Viola, Tubamaj, Tromcham, Travflot, Kleinerz, Corangla, Contreba, Clairmaj, Blockflo, Bassetho), with a total of 1306 recordings.

Organ: HuntsvilleAlabama

HuntsvilleAlabama (United States) - This sound bank represents 189 MB of sound data, consists of 4 registers (Swell, Pedal, Great, Choir), made of 25 stops (Vdgamb8, Vcele8, Stdiap8, Geigen8, Bourdon1, Twelfth, Trumpet8, Quintato, Princi8, Princi2, Octave4, Mixture, Hptrum8, Hohlfl8, Gedeckt8, Unda8, Tierce, Nazard, Nacht4, Mixture1, Gemsh4, Flhar8, Dulc8, Clarinet, Blockfl2), with a total of 446 recordings.

Organ: Huron

Huron (United States) - This sound bank represents 38 MB of sound data, consists of 1 register (Great), made of 14 stops (Trum8, Prin8, Prin4, Prin2, Prest4, Oboe, Nazard, Nasonfl, Kopflote, Gemshorn, Flautra4, Dagamba, Bourdon, Bombarde), with a total of 256 recordings.

Organ: KilgoreTexas

KilgoreTexas (United States) - This sound bank represents 111 MB of sound data, consists of 3 registers (Swell, Great, Choir), made of 26 stops (Voxhum, Viola, Trompett, Rohrflot, Principa, Nazard, Hautbois, Geigen, Flutetri, Flutedol, Flutebec, Clairon, Bombarde, Spitzflo, Bour8ve, Prinzipa, Oktav, Nasat, Montre, Kopflote, Harmsflt, Gedepomr, Gamba, Cromorne, Conflute, Basson), with a total of 535 recordings.

Organ: Knoxville

Knoxville (United States) - This sound bank represents 116 MB of sound data, consists of 1 register (Swell), made of 11 stops (Terz135, Spprinc4, Salizi8, Salcel8, Rohrfl8, Oboe8, Nasat223, Nachth4, Klarine4, Fagott16, Blockfl2), with a total of 222 recordings.

Organ: SacredHeart

SacredHeart (France) - This sound bank represents 114 MB of sound data, consists of 1 register (Misc), made of 7 stops (Violdo, Trumpet, Rohrflut, Opend, Musette, Hohlflte, Clarinet), with a total of 351 recordings.

Organ: Torrington

Torrington (England) - This sound bank represents 260 MB of sound data, consists of 4 registers (Swell, Pedal, Great, Choir), made of 31 stops (Voxhumana8tc, Salicional8, Principal4, Oboe8, Gedeckt8, Diapason8, Cornopean8, Contrafagotto16, Clarion4, Celeste8tc, Ophicleide16, Openwood16, Cello16, Principal 4', Posaune 8', Opendiapason2-8, Opendiapason16, Opendiapason1-8, Nasard, Mixture, Harmonicflute8, Fifteenth2, Clarion4, Violdamour, Stoppeddiapason8, Orchestraloboe8, Opendiapason8, Harmonicpiccolo2, Dulciana8, Cornodibassetto, Concertflute), with a total of 402 recordings.

Organ: Wicks

Wicks (United States) - This sound bank represents 258 MB of sound data, consists of 1 register (Misc), made of 30 stops (Violon bass, Violon, Tuba, swtrom8, swtro8s, swstfl8s, swstfl8, swsprn4, swspr4s, swspfl4s, swspfl4, swsal8s, swsal8, swnaz23s, swnaz223, swmix, swdul8, swbas16, pprn1684, ppos1684, pmixs, pdpos16, pdbou16s, pdbou16, gtprin8, Flasch8, Contrebour, Contrebass, Contrabomb, Basson), with a total of 478 recordings.

A.2 Rules for Fuzzy Model of Audio Expertise

Fuzzy rules defined with the help of the audio experts.

Rules dealing with Cluster issues

- **Rule 1** - IF (AttackTime IS VeryFast) AND (DecayTime IS Fast)
  THEN (Cluster IS Cluster10)
- **Rule 2** - IF (AttackTime IS Fast)
  THEN (Cluster IS Cluster11)
- **Rule 3** - IF (AttackTime IS Medium)
  THEN (Cluster IS Cluster12)
- **Rule 4** - IF (AttackTime IS Slow)
  THEN (Cluster IS Cluster20)
- **Rule 5** - IF (AttackTime IS VerySlow)
  THEN (Cluster IS Cluster21)
- **Rule 6** - IF (DecayTime IS Slow)
  THEN (Cluster IS Cluster40)
- **Rule 7** - IF (FreqMod IS High) OR (Noise IS High)
  THEN (Cluster IS Cluster50)
- **Rule 8** - IF (FreqMod IS Medium) OR (Noise IS VeryHigh)
  THEN (Cluster IS Cluster50)

Rules dealing with Attack issues

- **Rule 9** - IF (AmpMod IS Low) OR (AmpMod IS MedLow)
  THEN (AttackAmpModAmount IS Low) AND (AttackAmpModRate IS Low)

- **Rule 10** - IF (AmpMod IS Medium) OR (AmpMod IS HighMed)
  THEN (AttackAmpModAmount IS Medium) AND (AttackAmpModRate IS Medium)

- **Rule 11** - IF (FreqMod IS Low) OR (FreqMod IS MedLow) OR (AmpMod IS Medium)
  THEN (AttackFreqModAmount IS Low) AND (AttackFreqModRate IS Low)

- **Rule 12** - IF (FreqMod IS Medium) OR (AmpMod IS High)
  THEN (AttackFreqModAmount IS Medium) AND (AttackFreqModRate IS Medium)

- **Rule 13** - IF (FreqMod IS VeryLow)
  THEN (AttackFreqModAmount IS Medium) AND (AttackFreqModRate IS Medium)

Rules dealing with Sustain issues

- **Rule 14** - IF (AmpMod IS Low) OR (AmpMod IS MedLow)
  THEN (SustainAmpModAmount IS Low) AND (SustainAmpModRate IS Low)

- **Rule 15** - IF (AmpMod IS Medium) OR (AmpMod IS HighMed)
  THEN (SustainAmpModAmount IS Medium) AND (SustainAmpModRate IS Medium)

- **Rule 16** - IF (FreqMod IS Low) OR (AmpMod IS MedLow)
  THEN (SustainFreqModAmount IS Low) AND (SustainFreqModRate IS Low)

- **Rule 17** - IF (FreqMod IS Medium) OR (AmpMod IS High)
  THEN (SustainFreqModAmount IS Medium) AND (SustainFreqModRate IS Medium)

Rules dealing with Noise issues

- **Rule 18** - IF (Noise IS VeryLow) OR (Noise IS Low) OR (Noise IS Medium)
  THEN (NoiseAmount IS Low) AND (NoiseRate IS Medium)

- **Rule 19** - IF (Noise IS Medium) OR (Noise IS High) OR (Noise IS VeryHigh)
  THEN (NoiseAmount IS Medium) AND (NoiseRate IS High)

A.3 Table of Frequency / Keys

ADE	Key	Frequency	ADE	Key	Frequency	ADE	Key	Frequency
0	$C-1$	16.3 Hz	12	$C-0$	32.7 Hz	24	$C1$	65.4 Hz
1	$C\sharp-1$	17.3 Hz	13	$C\sharp-0$	34.6 Hz	25	$C\sharp1$	69.2 Hz
2	$D-1$	18.3 Hz	14	$D-0$	36.7 Hz	26	$D1$	73.4 Hz
3	$D\sharp-1$	19.4 Hz	15	$D\sharp-0$	38.8 Hz	27	$D\sharp1$	77.7 Hz
4	$E-1$	20.6 Hz	16	$E-0$	41.2 Hz	28	$E1$	82.4 Hz
5	$F-1$	21.8 Hz	17	$F-0$	43.6 Hz	29	$F1$	87.3 Hz
6	$F\sharp-1$	23.1 Hz	18	$F\sharp-0$	46.2 Hz	30	$F\sharp1$	92.4 Hz
7	$G-1$	24.4 Hz	29	$G-0$	48.9 Hz	31	$G1$	97.9 Hz
8	$G\sharp-1$	25.9 Hz	20	$G\sharp-0$	51.9 Hz	32	$G\sharp1$	103.8 Hz
9	$A-1$	27.5 Hz	21	$A-1$	55.0 Hz	33	$A1$	110 Hz
10	$A\sharp-1$	29.1 Hz	22	$A\sharp-1$	58.2 Hz	34	$A\sharp1$	116.5 Hz
11	$B-1$	30.8 Hz	23	$B-1$	61.7 Hz	35	$B1$	123.4 Hz
36	$C2$	130.8 Hz	48	$C3$	261.6 Hz	60	$C4$	523.2 Hz
37	$C\sharp2$	138.5 Hz	49	$C\sharp3$	277.1 Hz	61	$C\sharp4$	554.3 Hz
38	$D2$	146.8 Hz	50	$D3$	293.6 Hz	62	$D4$	587.3 Hz
39	$D\sharp2$	155.5 Hz	51	$D\sharp3$	311.1 Hz	63	$D\sharp4$	622.2 Hz
40	$E2$	164.8 Hz	52	$E3$	329.6 Hz	64	$E4$	659.2 Hz
41	$F2$	174.6 Hz	53	$F3$	349.2 Hz	65	$F4$	698.4 Hz
42	$F\sharp2$	184.9 Hz	54	$F\sharp3$	369.9 Hz	66	$F\sharp4$	739.9 Hz
43	$G2$	195.9 Hz	55	$G3$	391.9 Hz	67	$G4$	783.9 Hz
44	$G\sharp2$	207.6 Hz	56	$G\sharp3$	415.3 Hz	68	$G\sharp4$	830.6 Hz
45	$A2$	220.0 Hz	57	$A3$	440.0 Hz	69	$A4$	880.0 Hz
46	$A\sharp2$	233.0 Hz	58	$A\sharp3$	466.1 Hz	70	$A\sharp4$	932.3 Hz
47	$B2$	246.9 Hz	59	$B3$	493.8 Hz	71	$B4$	987.7 Hz
72	$C5$	1046.5 Hz	84	$C6$	2093.0 Hz	96	$C7$	4186.0 Hz
73	$C\sharp5$	1108.7 Hz	85	$C\sharp6$	2217.4 Hz	97	$C\sharp7$	4434.9 Hz
74	$D5$	1174.6 Hz	86	$D6$	2349.3 Hz	98	$D7$	4698.6 Hz
75	$D\sharp5$	1244.5 Hz	87	$D\sharp6$	2489.0 Hz	99	$D\sharp7$	4978.0 Hz
76	$E5$	1318.5 Hz	88	$E6$	2637.0 Hz	100	$E7$	5274.0 Hz
77	$F5$	1396.9 Hz	89	$F6$	2793.8 Hz	101	$F7$	5587.6 Hz
78	$F\sharp5$	1479.9 Hz	90	$F\sharp6$	2959.9 Hz	102	$F\sharp7$	5919.9 Hz
79	$G5$	1567.9 Hz	91	$G6$	3135.9 Hz	103	$G7$	6271.9 Hz
80	$G\sharp5$	1661.2 Hz	92	$G\sharp6$	3322.4 Hz	104	$G\sharp7$	6644.8 Hz
81	$A5$	1760.0 Hz	93	$A6$	3520.0 Hz	105	$A7$	7040.0 Hz
82	$A\sharp5$	1864.6 Hz	94	$A\sharp6$	3729.3 Hz	106	$A\sharp7$	7458.6 Hz
83	$B5$	1975.5 Hz	95	$B6$	3951.0 Hz	107	$B7$	7902.1 Hz
108	$C8$	8372.0 Hz						

Table A.1: Table for Musicom / Keys / Frequency

Listings

The MATLAB listings are the following:

- CatalogueCDROM and CatalogueOrgan, MATLAB scripts used to catalogue the sound database. CatalogueCDROM requires a path (strCDROM) which specify where the sound database is located on disk and automatically explores sound database to retrieve structures of pipe organ sound banks (organised as organs, registers, stops, waves files) and create reports in LaTeXformat with information such as stops names, total size on disk, like in Appendix .1. Diagram tree like in Figure 6.1 can also be generated automatically using similar scripts.

- InputsFuzzySets, OutputsFuzzySets and FuzzyRules are MATLAB scripts that define the fuzzy model of audio expertise described in Chapter 5. FuzzySystem is the MATLAB that implement the system linked with Audio features extracted by WinPVoc.

Note: Other MATLAB scripts can be obtained from the author by email request.

B.1 MATLAB listing - CatalogueCDROM

Listing B.1: Matlab listing to CDROM with sound banks

```
%
% (C) Brahim HAMADICHAREF - 1999-2003
%
% strPathCDROM is CDROM path

strCurrentPathMain = pwd;
addpath(strCurrentPathMain);

strORGANs = '';
ORGANs = 0;
TotalDataSizeMB = 0;

strCDROM = 'C:\Music\Waves\Musicom\'
% and look for folder with are Organs !

strCatalogueAll = [ strCDROM 'aOrgan_Report.tex' ];
fileCatAll = fopen(strCatalogueAll, 'w');
fprintf(fileCatAll, [ ' \n' ]);
fprintf(fileCatAll, [ 'Recordings are from famous Organs including ' ]);

cd(strCDROM);
DirOrgans = dir; % get file names
[y, iOrgan] = sort(lower({DirOrgans.name})); % get index
DirOrgans = DirOrgans(iOrgan); % sort the DirReg

% Huron, Texas, etc
% Start from 3 to avoid '.' and '..'
for nOrgan = 3:size(DirOrgans)
    if(DirOrgans(nOrgan).isdir == 1)
        % One more Register (only if they are Folders)
        ORGANs = ORGANs + 1;
        DirOrgans(nOrgan).name;

        strPathCDROM = [ strCDROM DirOrgans(nOrgan).name ];
        [strOrgan, DataSizeMB, Name_ReportTex] = CatalogueOrgan(strPathCDROM);
        TotalDataSizeMB = TotalDataSizeMB + DataSizeMB;
        fprintf(fileCatAll, [ strOrgan ' (' num2str(DataSizeMB) ' MBytes)' ]);
        if(nOrgan ~= size(DirOrgans))
            fprintf(fileCatAll, [ ', ' ]);
        end
    end
end

fprintf(fileCatAll, [ ' in total representing ' num2str(TotalDataSizeMB/1024) '
    GBytes of sound data. \n' ]);
fclose(fileCatAll);

ORGANs
dos('copy C:\Music\Waves\Musicom\*.tex C:\Latex\Thesis\CDROM\*.tex')

% Back wehere I was ...
cd(strCurrentPathMain);

dos('copy *.m C:\Latex\Thesis\Matlab\CDROM\*.m')
```

B.2 MATLAB listing - CatalogueOrgan

Listing B.2: Matlab listing to catalogue an Organ

```
function [strOrgan, DataSizeMB, Name_ReportTex] = CatalogueOrgan(PathCDROM)
% function CatalogueOrgan(strPathCDROM)
%
% (C) Brahim HAMADICHAREF - 1999-2003
%
% strPathCDROM is CDROM path

clc
%clear all

strCurrentPath = pwd;
% Back wehere I was ...
%cd(strCurrentPath);

REGISTERs = 0;
STOPs = 0;
WAVs = 0;
DataSize = 0;

strRegisters = '';
strStops = '';

strReport = '_Report';
strCatalogue = '_Catalogue';

if(nargin < 1)
    PathCDROM = [ 'C:\Music\Waves\Musicom\Wicks' ];
end

[Indices] = find(PathCDROM == '\');
strOrgan = PathCDROM(Indices(end)+1:length(PathCDROM))

Name_ReportTex = [strOrgan strReport];

strCatalogue = [ PathCDROM strCatalogue '.txt'];
if(exist(strCatalogue))
     delete(strCatalogue);
end

fileCat = fopen(strCatalogue, 'w');

cd(PathCDROM);
DirRegisters = dir; % get file names
[y, iRegister] = sort(lower({DirRegisters.name})); % get index
DirRegisters = DirRegisters(iRegister); % sort the DirReg

DirRegisters(2:size(DirRegisters)).name

% CHOIR, ECHO, GREAT, SWELL, etc
% Start from 3 to avoid '.' and '..'
for nRegister = 3:size(DirRegisters)
    if(DirRegisters(nRegister).isdir == 1)
        % One more Register (only if they are Folders)
        REGISTERs = REGISTERs + 1;
        strTmp = DirRegisters(nRegister).name;
        strTmp(2:end) = lower(strTmp(2:end));
        strRegisters = [ strTmp ', ' strRegisters ];
        %DirRegisters(nRegister).name;

        strPath = [PathCDROM '\' DirRegisters(nRegister).name];
        cd(strPath);
        DirStops = dir; % get file names
        [y, iStop] = sort(lower({DirStops.name})); % get index
        DirStops = DirStops(iStop); % sort the DirReg

        % BASSON, VIOLA, etc
        % Start from 3 to avoid '.' and '..'
        for nStop = 3:size(DirStops)
            if((DirStops(nStop).name(1)) ~= ['.'])
                % One more Register (only if they are Folders)
```

```
71                  STOPs = STOPs + 1;
72
73                  strTmp = DirStops(nStop).name;
74                  strTmp(2:end) = lower(strTmp(2:end));
75                  strStops = [ strTmp ', ' strStops ];
76                  %strStops = [ DirStops(nStop).name ', ' strStops ];
77                  %DirStops(nStop).name;
78
79                  strPath = [PathCDROM '\' DirRegisters(nRegister).name '\' DirStops(
                        nStop).name];
80                  cd(strPath);
81                  DirWaves = dir; % get file names
82                  [y, iWAV] = sort(lower({DirWaves.name})); % get index
83                  DirWaves = DirWaves(iWAV); % sort the DirReg
84
85                  % 24., 25, etc
86                  % Start from 3 to avoid '.' and '..'
87                  for nWav = 3:size(DirWaves)
88                      if((DirWaves(nWav).isdir) == 1)
89                          disp('Error ... No expecting !');
90                      else
91                          if(DirWaves(nWav).name(1:5) ~= 'SLICE')
92                              WAVs = WAVs + 1;
93                              DataSize = DataSize + DirWaves(nWav).bytes;
94                              %DirWaves(nWav).name;
95                              strPathWAV = [PathCDROM '\' DirRegisters(nRegister).name
                                    '\' DirStops(nStop).name '\' DirWaves(nWav).name];
96                              fprintf(fileCat, '%s\n', strPathWAV(1:end));
97                          end
98                      end
99                  end
100             end
101         end
102     end
103 end
104 fclose(fileCat);
105
106 % Generate Latex report
107 strReportLatex = [ PathCDROM strReport '.tex' ];
108 if(exist(strReportLatex))
109      delete(strReportLatex);
110 end
111
112 fileTex= fopen(strReportLatex, 'w');
113 fprintf(fileTex, ['\\subsection*{Organ: ' strOrgan '} \n' ]);
114 fprintf(fileTex, [' \n' ]);
115 fprintf(fileTex, ['\\textbf{' strOrgan '} (United States / England / Canada / France
        / Germany) - ' ]);
116 %fprintf(file, ['%% ' num2str(REGISTERs) ' REGISTERs ' num2str(STOPs) ' STOPs '
        num2str(WAVs) ' WAVs \n' ]);
117 DataSizeMB = floor((DataSize/1024)/1024);
118 fprintf(fileTex, ['This sound bank represents ' num2str(DataSizeMB) ' MB of sound
        data']);
119 if(REGISTERs == 1)
120     fprintf(fileTex, [', consists of ' num2str(REGISTERs) ' register (' strRegisters
            (1:end-2) ')']);
121 else
122     fprintf(fileTex, [', consists of ' num2str(REGISTERs) ' registers (' strRegisters
            (1:end-2) ')']);
123 end
124
125 if(STOPs == 1)
126     fprintf(fileTex, [', made of ' num2str(STOPs) ' stops (' strStops(1:end-2) ')']);
127 else
128     fprintf(fileTex, [', made of ' num2str(STOPs) ' stops (' strStops(1:end-2) ')']);
129 end
130 fprintf(fileTex, [', with a total of ' num2str(WAVs) ' recordings. \n' ]);
131 fclose(fileTex);
132
133 % Only for example in thesis
134 %
135 % requires \usepackage{pstricks}, \usepackage{pst-node} and \usepackage{pst-tree}
```

```
% and few macros
%
%\def\Tbox#1{\Tr{\psframebox[framearc=.5]{\centered{#1}}}}
% \def\arrowbit{%
%   \cnode(0,0){.1}{A}\pnode(0.6,0){B}\ncline{->}{A}{B}
%   }
% \def\luarrow#1{\nbput[nrot=:D]{\arrowbit\rput[br]{*0}(-0.1,0.05){\small#1}}}
% \def\ruarrow#1{\naput[nrot=:D]{\arrowbit\rput[bl]{*0}(-0.1,0.05){\small#1}}}
% \def\ldarrow#1{\nbput[nrot=:U]{\arrowbit\rput[br]{*0}(-0.1,0.05){\small#1}}}
% \def\rdarrow#1{\naput[nrot=:U]{\arrowbit\rput[bl]{*0}(-0.1,0.05){\small#1}}}

% CreateTreeDiagram(PathCDROM)

% Back wehere I was ...
cd(strCurrentPath);
```

B.3 MATLAB listing - InputsFuzzySets

Listing B.3: Create all 11 Inputs of fuzzy model

```
%   - Fuzzy Model of Modeling Audio Expertise -
%
% Brahim HAMADICHAREF (C) 1999-2003

clc
clear all
close all

epsDPI = '-r300';
jpegDPI = '-r90';

fis = newfis('AudioExpert05');

ni = 0;
no = 0;
N = 2.25;

FSetVal = [-7 3 12 30 80 100];
mfStr = ['VeryFast';'Fast    ';'Medium  ';'Slow    ';'VerySlow'];
[fis, ni] = a_CreateInput(fis, ni, 'AttackTime', mfStr, FSetVal, 'sigmf', N);
[fis, ni] = a_CreateInput(fis, ni, 'ReleaseTime', mfStr, FSetVal);

FSetVal = [-10 5 15 100];
mfStr = ['Fast  ';'Medium';'Slow  '];
[fis, ni] = a_CreateInput(fis, ni, 'DecayTime', mfStr, FSetVal);

FSetVal = [-8.33 8.33 16.66 25 33.33 41.65 50 58.31 66.64 75 83.3 91.63 100];
mfStr = ['VeryLow  ';'Low      ';'Low2     ';'LowMed   ';'Medium   ';
         'HighMed  ';'High2    ';'High3    ';'VeryHigh ';'VeryHigh2';
         'VeryHigh3';'Noisy    '];
[fis, ni] = a_CreateInput(fis, ni, 'Harmonic', mfStr, FSetVal);

FSetVal = [-5 5 25 50 70 85 95 100];
mfStr = ['VeryLow  ';'Low      ';'LowMed   ';'Medium   ';
         'HighMed  ';'High     ';'VeryHigh '];
[fis, ni] = a_CreateInput(fis, ni, 'Amplitude', mfStr, FSetVal);

FSetVal = [-10 10 20 35 50 100];
mfStr = ['VeryLow  ';'Low      ';'Medium   ';
         'High     ';'VeryHigh '];
[fis, ni] = a_CreateInput(fis, ni, 'Frequency', mfStr, FSetVal);

FSetVal = [-5 5 15 30 50 75 100];
mfStr = ['VeryLow  ';'Low      ';'MedLow   ';
```

```
45          'Medium    ';'High      ';'VeryHigh ';
46  [fis, ni] = a_CreateInput(fis, ni, 'AmpMod', mfStr, FSetVal);
47
48  FSetVal = [-6 6 12 24 49 95 100];
49  mfStr = ['VeryLow   ';'Low       ';'MedLow    ';
50           'Medium    ';'High      ';'VeryHigh ';
51  [fis, ni] = a_CreateInput(fis, ni, 'FreqMod', mfStr, FSetVal);
52
53  %FSetVal = [-10 26 22 38 34 50 46 62 58 74 70 86 82 100];
54  %mfStr = ['VeryLow   ';'Low       ';'MedLow    ';'Medium    ';
55  %          'MedHigh  ';'High      ';'VeryHigh ';
56  %[fis, ni] = a_CreateInput(fis, ni, 'Pitch', mfStr, FSetVal);
57
58  FSetVal = [-10 20 40 60 80 100];
59  mfStr = ['VeryLow   ';'Low       ';
60           'Medium    ';'High      ';'VeryHigh ';
61  [fis, ni] = a_CreateInput(fis, ni, 'Brightness', mfStr, FSetVal);
62
63  FSetVal = [-10 20 40 60 80 100];
64  mfStr = ['VeryLow   ';'Low       ';
65           'Medium    ';'High      ';'VeryHigh ';
66  [fis, ni] = a_CreateInput(fis, ni, 'Noise', mfStr, FSetVal);
67
68  %% Save eps/m to LaTeX thesis folder %%%%%%%%%%%%%%%%%%%%%%%%%
69  dos('copy *.eps C:\Latex\Thesis\Figures\FuzzyExpertSystem\*.eps')
70  dos('copy *.m C:\Latex\Thesis\Matlab\FuzzyExpertSystem\*.m')
```

Listing B.4: Create one Input fuzzy variable

```
1   %   - Fuzzy Model of Modeling Audio Expertise -
2   %
3   % Brahim HAMADICHAREF (C) 1999-2003
4
5   function [fis, ni] = a_CreateInput(fis, ni, nameStr, mfStr, FSetVal, MemFuncStr, N)
6
7   if(nargin < 7)
8       N = 2.25;
9   end
10  if(nargin < 6)
11      N = 2.25;
12      MemFuncStr = 'sigmf';
13  end
14
15  fis = addvar(fis,'input',nameStr,[0 FSetVal(end)]);
16  ni = ni + 1;
17  for k =1:size(mfStr, 1)-1
18      fis = addmf(fis,'input',ni,strcat(mfStr(k,1:end)),['p' MemFuncStr(:)'],[-N
            FSetVal(k+1) N FSetVal(k)]);
19  end
20  fis = addmf(fis,'input',ni,strcat(mfStr(end,1:end)),MemFuncStr(:)',[N FSetVal(end-1)
        ]);
21  figure(ni)
22  set(gca, 'FontSize', 14);
23  plotmf(fis,'input',ni)
24  title(['Fuzzy Input - ' nameStr])
25  grid on
26  %hLine = get(gca,'Children');
27  %set(hLine,'Color', 'k','LineWidth',2);
28  %set(findobj(gca,'type','text'), 'FontSize', 14);
29
30  MaximizeWnd
31  Wygiwys
32  set(gca, 'Position', [0.210 0.100 0.775 0.815]);
33  if (exist('epsDPI','var') == 0), epsDPI = '-r300'; end
34  if (exist('jpegDPI','var') == 0), jpegDPI = '-r90'; end
35  print('-deps', epsDPI, ['Fig_FSet_' nameStr])
36  print('-djpeg', jpegDPI, ['Fig_FSet_' nameStr])
```

B.4 MATLAB listing - OutputFuzzySets

Listing B.5: Create all 15 Outputs of fuzzy model

```
%   - Fuzzy Model of Modeling Audio Expertise -
%
% Brahim HAMADICHAREF (C) 1999-2003

clc
clear all
close all

epsDPI = '-r300';
jpegDPI = '-r90';

fis = newfis('AudioExpert05');

ni = 0;
no = 0;
N = 2.25;

FSetVal = [-7 3 12 30 80 100];
mfStr = ['VeryFast';'Fast    ';'Medium  ';'Slow    ';'VerySlow'];
[fis, ni] = a_CreateInput(fis, ni, 'AttackTime', mfStr, FSetVal, 'sigmf', N);

FSetVal = [-10 10 20 30 40 50 60 70 80 90 100];
mfStr = [ 'Cluster10';'Cluster11';'Cluster12';
          'Cluster20';'Cluster21';'Cluster22';
          'Cluster30';'Cluster31';
          'Cluster40';'Cluster50'];
[fis, no] = a_CreateOutput(fis, no, 'Cluster', mfStr, FSetVal, 'sigmf', N);

%%%%%%%%%%%%%%%%%%%%%%%% Output - Attack %%%%%%%%%%%%%%%%%%%%%%%%
% Amp, Freq, AmpModAmount, AmpModRate, FreqModAmount, FreqModRate)

FSetVal = [-10 15 40 60 82 100];
mfStr = [ 'VeryLow ';'Low     ';'High    ';'Medium  ';'VeryHigh'];
[fis, no] = a_CreateOutput(fis, no, 'AttackAmp', mfStr, FSetVal, 'sigmf', N);

FSetVal = [-10 13 43 60 80 100];
mfStr = [ 'VeryLow ';'Low     ';'High    ';'Medium  ';'VeryHigh'];
[fis, no] = a_CreateOutput(fis, no, 'AttackFreq', mfStr, FSetVal, 'sigmf', N);

FSetVal = [-10 21 39 52 82 100];
mfStr = [ 'VeryLow ';'Low     ';'High    ';'Medium  ';'VeryHigh'];
[fis, no] = a_CreateOutput(fis, no, 'AttackAmpModRate', mfStr, FSetVal, 'sigmf', N);

FSetVal = [-10 16 40 60 76 100];
mfStr = [ 'VeryLow ';'Low     ';'High    ';'Medium  ';'VeryHigh'];
[fis, no] = a_CreateOutput(fis, no, 'AttackAmpModAmount', mfStr, FSetVal, 'sigmf', N
    );

FSetVal = [-10 18 41 60 82 100];
mfStr = [ 'VeryLow ';'Low     ';'High    ';'Medium  ';'VeryHigh'];
[fis, no] = a_CreateOutput(fis, no, 'AttackFreqModRate', mfStr, FSetVal, 'sigmf', N);

FSetVal = [-10 23 38 61 81 100];
mfStr = [ 'VeryLow ';'Low     ';'High    ';'Medium  ';'VeryHigh'];
[fis, no] = a_CreateOutput(fis, no, 'AttackFreqModAmount', mfStr, FSetVal, 'sigmf', N
    );

%%%%%%%%%%%%%%%%%%%%%%%% Output - Sustain %%%%%%%%%%%%%%%%%%%%%%%%
% Amp, Freq, AmpModAmount, AmpModRate, FreqModAmount, FreqModRate)

FSetVal = [-10 22 43 62 80 100];
mfStr = [ 'VeryLow ';'Low     ';'High    ';'Medium  ';'VeryHigh'];
[fis, no] = a_CreateOutput(fis, no, 'SustainAmp', mfStr, FSetVal, 'sigmf', N);

FSetVal = [-10 25 40 61 79 100];
mfStr = [ 'VeryLow ';'Low     ';'High    ';'Medium  ';'VeryHigh'];
```

```
[fis, no] = a_CreateOutput(fis, no, 'SustainFreq', mfStr, FSetVal, 'sigmf', N);

FSetVal = [-10 18 42 60 81 100];
mfStr = [ 'VeryLow ';'Low     ';'High    ';'Medium  ';'VeryHigh'];
[fis, no] = a_CreateOutput(fis, no, 'SustainAmpModRate', mfStr, FSetVal, 'sigmf', N);

FSetVal = [-10 20 40 62 80 100];
mfStr = [ 'VeryLow ';'Low     ';'High    ';'Medium  ';'VeryHigh'];
[fis, no] = a_CreateOutput(fis, no, 'SustainAmpModAmount', mfStr, FSetVal, 'sigmf', N
    );

FSetVal = [-10 23 37 60 79 100];
mfStr = [ 'VeryLow ';'Low     ';'High    ';'Medium  ';'VeryHigh'];
[fis, no] = a_CreateOutput(fis, no, 'SustainFreqModRate', mfStr, FSetVal, 'sigmf', N)
    ;

FSetVal = [-10 20 38 59 80 100];
mfStr = [ 'VeryLow ';'Low     ';'High    ';'Medium  ';'VeryHigh'];
[fis, no] = a_CreateOutput(fis, no, 'SustainFreqModAmount', mfStr, FSetVal, 'sigmf',
    N);

%%%%%%%%%%%%%%%%%%%%%%%% Noise Generator %%%%%%%%%%%%%%%%%%%%%%%%
% NoiseAmount and NoiseRate

FSetVal = [-10 31 45 60 75 100];
mfStr = [ 'VeryLow ';'Low     ';'High    ';'Medium  ';'VeryHigh'];
[fis, no] = a_CreateOutput(fis, no, 'NoiseAmount', mfStr, FSetVal, 'sigmf', N);

FSetVal = [-10 34 50 60 78 100];
mfStr = [ 'VeryLow ';'Low     ';'High    ';'Medium  ';'VeryHigh'];
[fis, no] = a_CreateOutput(fis, no, 'NoiseRate', mfStr, FSetVal, 'sigmf', N);

dos('copy *.eps C:\Latex\Thesis\Figures\FuzzyExpertSystem\*.eps')
dos('copy *.m C:\Latex\Thesis\Matlab\FuzzyExpertSystem\*.m')

close all
```

Listing B.6: Create one Output fuzzy variable

```
%    - Fuzzy Model of Modeling Audio Expertise -
%
% Brahim HAMADICHAREF (C) 1999-2003

function [fis, ni] = a_CreateInput(fis, ni, nameStr, mfStr, FSetVal, MemFuncStr, N)

if(nargin < 7)
    N = 2.25;
end
if(nargin < 6)
    N = 2.25;
    MemFuncStr = 'sigmf';
end

fis = addvar(fis,'output',nameStr,[0 FSetVal(end)]);
ni = ni + 1;
for k =1:size(mfStr, 1)-1
    fis = addmf(fis,'output',ni,strcat(mfStr(k,1:end)),['p' MemFuncStr(:)'],[-N
        FSetVal(k+1) N FSetVal(k)]);
end
fis = addmf(fis,'output',ni,strcat(mfStr(end,1:end)),MemFuncStr(:)',[N FSetVal(end-1)
    ]);
figure(ni)
set(gca, 'FontSize', 14);
plotmf(fis,'output',ni)
title(['Fuzzy Output - ' nameStr])
grid on
%hLine = get(gca,'Children');
%set(hLine,'Color', 'k','LineWidth',2);
%set(findobj(gca,'type','text'), 'FontSize', 14);

```

```
MaximizeWnd
Wygiwys
set(gca, 'Position', [0.210 0.100 0.775 0.815]);
if (exist('epsDPI','var') == 0), epsDPI = '-r300'; end
if (exist('jpegDPI','var') == 0), jpegDPI = '-r90'; end
print('-deps', epsDPI, ['Fig_FSet_' nameStr])
print('-djpeg', jpegDPI, ['Fig_FSet_' nameStr])
```

B.5 MATLAB listing - FuzzyRules

Listing B.7: Code to create fuzzy rules

```
% function a_create_rules
%
% Fuzzy Exert System to Sound Modeling
%   - Modeling Audio Expertise -
%
% Brahim HAMADICHAREF (C) 1999-2003

%%%%%%%%%%%%%%%%%%%%%%%%% Rules  %%%%%%%%%%%%%%%%%%%%%%%%%

%
% See whole system
% MyShowRules(a,1:size(a.rule, 2))
% showrule(a,1:size(a.rule, 2),'symbolic')
%
% 11 inputs and 15 outputs
%
% I - inputs / O - Outputs
% W - Weight / B - binary AND/OR

ruleList=[

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Note
%   I      Inputs     ... I O ...         Outputs      ... O W B
%   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1;
%
% Cluster10
%   I                    I O ...                      ... O W B
    1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1;
% Cluster11
    2 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1;
% Cluster20
    3 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1;
    4 0 0 0 0 0 0 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1;
    5 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1;
% Cluster21
%   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1;
% Cluster22
%   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1;
% Cluster31
%   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1;
% Cluster31
%   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1;
% Cluster32
%   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1;
% Cluster40
%   0 0 0 0 0 0 0 0 0 0 0 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Cluster50 ... If Noise or FreqMod are High or VeryHigh
    0 0 0 0 0 0 0 5 0 0 4 10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2;
    0 0 0 0 0 0 0 4 0 0 5 10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2;
    0 0 0 0 0 0 0 4 0 0 0 10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
];
a = addrule(a, ruleList);

a_ShowRules(a, 1:size(a.rule,2));
dos('copy *.tex C:\Latex\Thesis\Matlab\FuzzyExpertSystem\*.tex')
```

B.6 MATLAB listing - FuzzySystem

Listing B.8: Code to use the fuzzy system

```
%function audio_expertise06

% Fuzzy Exert System to Sound Modeling
%   - Modeling Audio Expertise -
%
% Brahim HAMADICHAREF (C) 1999-2003

clc
close all
clear all
warning off;
format long;

a = newfis('AudioExpert06');

epsDPI = '-r300';
jpegDPI = '-r90';

ni = 0;
no = 0;
N = 2.25;

nMax = 100;

a_create_inputs %(a, ni, no, N);
close all

a_create_outputs %(a, ni, no, N);
close all

a_create_rules

% Get resusults from analysis for the fuzzy inputs
[InputAttackTime, InputDecayTime, InputReleaseTime, ...
    InputHarmonic, InputAmplitude, InputFrequency, ...
    InputAmpMod, InputFreqMod, InputPitch, ...
    InputBrightness, InputNoise] = a_InputsAnalysis(nMax);

Inputs = [  InputAttackTime;
            InputDecayTime;
            InputReleaseTime;
            InputHarmonic;
            InputAmplitude;
            InputFrequency;
            InputAmpMod;
            InputFreqMod;
            InputPitch;
            InputBrightness;
            InputNoise ];

% Evaluate Fuzzy model of Audio Expertise
[Outputs] = evalfis(Inputs, a);
```

```
53
54 % plot outputs
55
56 NbOutputs = size(Outputs, 2);
57
58 for nOutput = 1:NbOutputs
59     figure(100 + nOutput)
60     set(gca, 'FontSize', 14);
61     plot(Outputs(:,nOutput), 'ko-')
62     strTitle = [a.output(nOutput).name];
63     FigStr = [ 'Fig_Output' strTitle];
64     title(strTitle);
65     xlabel('Time');
66     grid on
67     axis([1 nMax 0 100])
68     MaximizeWnd
69     Wygiwys
70     print('-deps', epsDPI, FigStr)
71     print('-djpeg', jpegDPI, FigStr)
72 end
73
74 figure(101)
75 set(gca,'YTickLabel','Cluster10|Cluster11|Cluster12|Cluster20|Cluster21|Cluster22|
       Cluster30|Cluster31|Cluster40|Cluster50');
76
77 % a.input(1:size(a.input, 2)).name
78 % a.output(1:3).name
79
80 showrule(a,1:size(a.rule, 2),'indexed')
81 %showrule(a,1:size(a.rule, 2),'symbolic');
82
83 dos('copy *.eps C:\Latex\Thesis\Figures\FuzzyExpertSystem\*.eps')
84 dos('copy *.m C:\Latex\Thesis\Matlab\FuzzyExpertSystem\*.m')
85
86 % close all
87
88 % showfis(a)
```

Publications

C.1 Paper 1 - DAFX99

Hamadicharef, B. and Ifeachor, E. C. "Artificial Intelligence based Modeling of Musical Instruments", in Proceedings of the 2nd COST-G6 Workshop on Digital Audio Effects December 9-11, 1999, Trondheim, Norway.

Abstract

In this paper, a novel research tool, which allows real-time implementation and evaluation of sound synthesis of musical instrument, is described. The tool is a PC-based application and allows the user to evaluate the effects of parameter changes on the sound quality in an intuitive manner. Tuning makes use of a Genetic Algorithm (GA) technique. Flute and plucked string modeling examples are used to illustrate the capabilities of the tool.

Proceedings of the 2nd COST G-6 Workshop on Digital Audio Effects (DAFx99)
NTNU, Trondheim, December 9-11, 1999

ARTIFICIAL INTELLIGENCE BASED MODELING OF MUSICAL INSTRUMENTS

Brahim Hamadi Charef, Emmanuel Ifeachor

SMART Systems Research Group
School of Electronics, Communication and Electrical Engineering
University of Plymouth
Drake Circus, Plymouth PL4 8AA, Devon, England
brahim@cis.plym.ac.uk ; E.Ifeachor@plymouth.ac.uk

ABSTRACT

In this paper, a novel research tool, which allows real-time implementation and evaluation of sound synthesis of musical instrument, is described. The tool is a PC-based application and allows the user to evaluate the effects of parameter changes on the sound quality in an intuitive manner. Tuning makes use of a Genetic Algorithm (GA) technique. Flute and plucked string modeling examples are used to illustrate the capabilities of the tool.

1. INTRODUCTION

Over the last decade, sound synthesis, especially physical modeling of musical instruments, has emerged as an important research field in computer music. Physical modeling, which refers to the computational simulation of the acoustics of musical instruments, is the most promising sound synthesis technique at present [1].

Two major problems in computer modeling of musical instruments are: (1) the difficulty in producing realistic sound and (2) the challenge of finding the optimal parameters of the computer model in order to produce the desired sound. This paper is concerned with finding a solution to the above problems.

The work presents the development of a software environment for the study and exploration of computer modeling of plucked, bowed and blown musical instruments, with particular interest in wind instruments, such as flute and organ pipe, using digital waveguide models [1][2]. The user-friendly, PC-based research tool offers the user the possibility to explore the inter-relationship between the instrument sound and the model parameters. An important requirement is to allow the user to evaluate the instrument model quickly in real-time for an immediate assessment of the effects of parameter changes on sound quality. The tool also allows accurate tuning of the model parameters based on recordings of real instruments using an intelligent algorithm.

2. SOUND SYNTHESIS SYSTEM

The sound synthesis system is based on a multimedia computer. The computer should be capable of sustaining sound synthesis in real-time. In the current system, the computer is a PC (Pentium II 450 MHz) running under Windows NT 4.0 operating system. The interaction is done by the use of the mouse and keyboard. Later MIDI control will be added to allow performance with various MIDI controllers such as keyboards and electronic wind controller (Yamaha WX11).

3. SYSTEM SOFTWARE

For this research tool, two software libraries have been developed using Visual C++ 6.0. The first library is a set of Audio Elements, the second library is made of Control Elements, and each of them is associated with an audio element. The tools developed make extensive use of these libraries. Each consists of a Sound Engine for sound synthesis and a Control Engine to give the user full access to the instrument model.

3.1. Audio Library and Sound Engine

The audio library consists of Digital Signal Processing (DSP) building blocks commonly found in audio including noise generators (white, pink, etc.) multi-segment envelope generators (classic attack-decay-sustain-release) frequency selective filters (e.g. low-pass, high-pass, band-pass, stop-band and DC killer). The audio library also contains interpolating (linear and higher order Allpass and Lagrange filters [3]) and polynomials. The Sound Engine is a real-time thread that implements the model of the musical instrument. It can sustain real-time sound synthesis giving the user immediate results of the capabilities of the model. The Sound Engine has also a real-time input that can be used as an excitation signal for the model. Finally it can generate WAV and AIFF files that can be used by any standard sound editing software.

3.2. Control Library and Control Engine

Each audio element of the audio library has its associated control element. This makes use of the Windows graphic objects such as sliders and other custom controls. The Graphic User Interface (GUI) shows the shape of the musical instrument to be modeled can be visualized. Each part of the instrument model has associated control elements, which give the user access to sliders to freely adjust the model parameters.

3.3. Intelligent Optimiser

To accurately tune the model an intelligent algorithm has been used. Following the guidelines from [4] we have added a feature to allow tuning from real recording of organ pipe and flute instruments.

The population is first randomly initialised and then evolves towards better solutions through a selection process, i.e. the selection of the best individuals based on their fitness score, a mutation process introducing some innovation into the population and recombination process for the next population. The fitness value is determined by the similarity of the harmonics peaks of the sound generated by the model to those of the recording of the acoustic instrument.

4. APPLICATION EXAMPLES

At present, two models of musical instruments have been implemented using the sound synthesis system and these are described below.

4.1. Flute Model

A flute model based on Cook's air-jet pipe model [5] and Välimäki's improved flute model [6] has been implemented. This model consists of an excitation, a non-linear and a linear part. The excitation, modeling the action of the musician on the instrument, uses filtered noise with an envelope generator. The non-linear interaction between the air-jet and the tube resonator is modeled by polynomial. The linear part, the resonator, is built using a network of multi-section tube and tone holes junctions. A dedicated GUI called the Pipe Modeler was developed for this flute model.

4.1.1. Pipe Modeler GUI

The Pipe Modeler is a multi property page application dialog based application. It consists of the Excitation, the Non-linear, the Linear, the Waveform, the Spectrum, the Output and the File pages.

4.1.2. Excitation Page

The excitation of the flute model uses a low pass filtered noise generator with an Attack-Decay-Sustain-Release (ADSR) amplitude envelope generator. The control elements for the noise generator, the digital filter and ADSR envelope generator have been developed. The user can adjust the noise level, cutoff frequency of low-pass filter, and time constants of the ADSR envelope (in ms). Various ADSR envelope characteristics can be used to emulate a variety of the flute features from a pulse to very slow raising blowing characteristics.

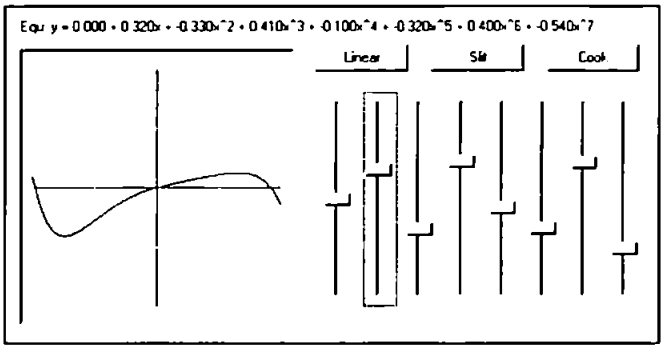


Figure 1. *Control element of the polynomial.*

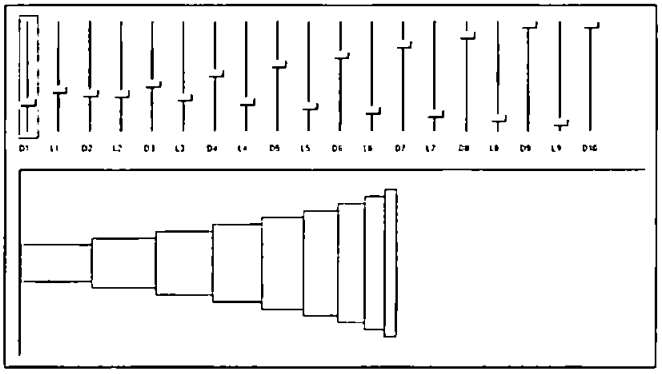


Figure 2. *Control element of the multi-sections tube*

4.1.3. Non-linear Page

The main element of the non-linear page is the polynomial transfer function, which is used to model the air-jet of the flute. The user can use the sliders to adjust the corresponding polynomial coefficients. The polynomial curve is automatically updated. Figure 1 shows the control element for the polynomial function. Presets can be assigned to three buttons (for example: linear, slit and Cook characteristic [5]).

4.1.4. Linear Page

With the linear page, the user can easily edit the shape of the pipe resonator of the model. In the Pipe Modeler, the pipe is made of several subsections of cylindrical tubes and tone hole junctions. For each subsection the diameter and the length can be adjusted, allowing the user to experiment with the shape of the tube. An example of control element for the multi-section tube is shown in Figure 2.

4.1.5. Waveform Page

The waveform page gives a time domain representation of the sound produced by the model. Any part of the produced sound can be zoomed. The attack and sustain parts of the produced sound are interesting. Visual clues about the quality of the sound can be obtained from inspection of the waveform of the produced sound.

4.1.6. Spectrum Page

A frequency representation of the produced sound is shown in the spectrum page of the tool. The user can choose between a linear and logarithmic scale. Visual clues about the tone color of the sound can also be obtained from inspection of the spectrum. The spectrum is also used by the Genetic Optimizer to calculate the fitness function.

4.1.7. Output Page

The output page gives the opportunity to the user to specify which signals of the model to be used and mixed and by which amount to produce the final sound output. This feature has also been found useful during the testing of the different audio elements of the model. The same way, the real-time input can be injected in any part of the model. Figure 3.a shows the waveform of synthesized flute sound and Figure 3.b. its 3D spectrum.

4.1.8. File Page

The File page is concerned with disk operations to save and load configuration files. After experimenting with the model, the user can save the configuration of the Pipe Modeler that gives the best results. Finally, the File page allows selecting the WAV file template (acoustic recording) used for the calculation of the fitness value for the Genetic Optimiser.

4.3. Plucked String Model

The plucked string model we used is a simple computational technique for modeling plucked string sounds based on refinements of the Karplus-Strong (KS) algorithm [7]. The model has been improved over the original version using a 3rd order Lagrange interpolator fractional delay filter to fine-tune the pitch and a low-pass filter (also called loop filter) to model the frequency-dependent damping of the string as described in [8]. The model is runs at a sampling rate of 44.1 kHz and is excited (pluck action of the musician) using various filtered bursts of noise to obtain different tone color plucked string sound. This plucked string model, despite its simplicity, produces realistic and high quality sounds. The waveform of synthesized plucked string tone and its 3D spectrum are shown in Figure 4.a and 4.b. respectively. The plucked string model will be demonstrated at the conference.

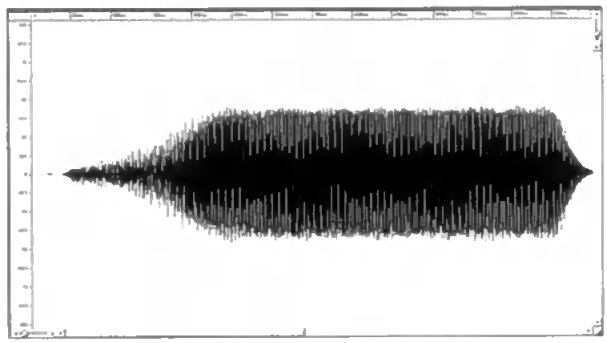


Figure 3.a. *Flute synthesised tone.*

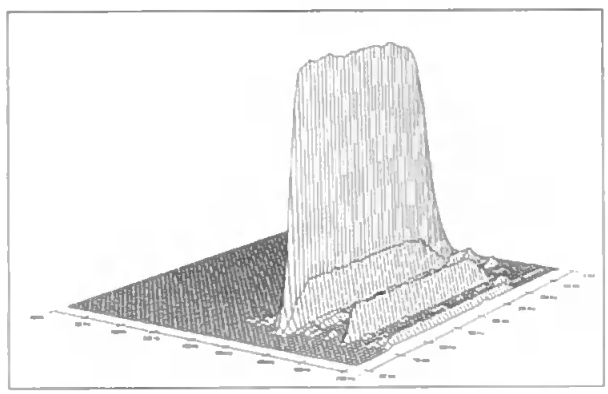


Figure 3.b. *3D Spectrum of the flute synthesised tone.*

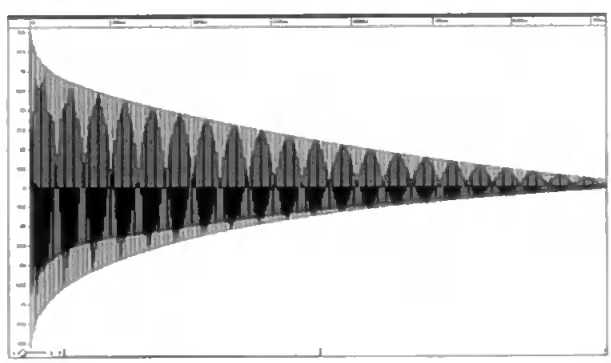


Figure 4.a. *Plucked String synthesised tone.*

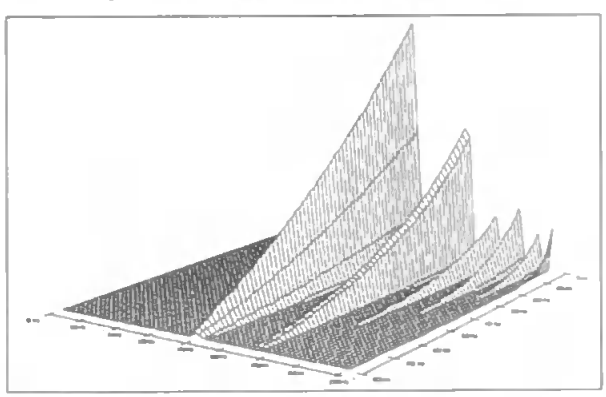


Figure 4.b. *3D spectrum of the plucked string synthesised tone.*

5. CONCLUSIONS AND FUTURE WORK

In this work, a computer-based environment for real-time implementation and evaluation of sound synthesis of musical instruments has been presented. The main motivation behind it is to develop a flexible environment that allows the user to experiment with the instrument models, using the real-time capabilities to listen to the sound produced. We have illustrated the potential of the sound synthesis environment by examples including flute and plucked string modeling.

In future, the library of audio elements for the sound engine and their associated control elements will be extended. Other sound synthesis techniques such as additive and Frequency Modulation (FM) will also be added. Control issues will also be considered using an electronic wind controller. Finally, the real-time input feature will allow real excitation signals to be used to drive the models.

This will provide the basis for an advanced investigation into Artificial Intelligence (AI) based modeling of audio expertise that will help to improve the modeling of musical instruments and control of the sound synthesis.

6. REFERENCES

[1] Smith, J. O., "Physical Modeling Synthesis Update", *Computer Music J., 20(2): 44-56, 1996.*

[2] Smith, J. O., "Physical modeling using digital waveguides", *Computer Music J., 16(4): 74-87, 1992.*

[3] Laakso, T. I., Välimäki, V., Karjalainen, M. and Laine, U. K., "Crushing the Delay - Tools for Fractional Delay Filter Design", Report no. 35, Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing, 1994.

[4] Vuori, J. and Välimäki, V., "Parameter Estimation of non-linear physical models by simulated evolution – application to the flute model", *Proc. Int. Computer Music Conf. (ICMC'93), : 402-404, Tokyo, Japan, Sept. 10-15, 1993.*

[5] Cook, P. R. "A Meta-Wind-Instrument Physical Model and a Meta-Controller for real Time Performance Control", *Proc. Int. Computer Music Conf. (ICMC'92), : 273-276, San Jose, California, Oct. 14-18, 1992.*

[6] Hänninen, R. and Välimäki, V., "An improved digital waveguide model of a flute with fractional delay filters", *Proc. Nordic Acoustical Meeting (NAM'96), : 437-444, Helsinki, Finland, June 12-14, 1996.*

[7] K. Karplus, and A. Strong, "Digital synthesis of plucked-string and drum timbres", *Computer Music Journal*, 7(2): 42-55, 1983.

[8] Välimäki, V., and Tolonen, T., "Development and calibration of a guitar synthesizer", *J. Audio Eng. Soc.* 46(9): 766-778, 1998.

C.2 Paper 2 - AES111

Hamadicharef, B. and Ifeachor, E. C. "An Intelligent System Approach to Sound Synthesis Parameter Optimisation", in Proceedings of the 111^{th} Audio Engineering Society convention, November 30 - December 3, 2001, New York, USA. Preprint 5484.

Abstract

An intelligent audio system for sound design using artificial intelligence techniques is reported. The system is used to analyse acoustic recordings, extract salient sound features and to process them to generate parameters for sound synthesis, in a manner that mimics human audio experts. Preliminary tests show that the use of the system reduces design time and yet the quality of the resulting sound is considered high by audio experts.

Audio Engineering Society

Convention Paper 5484

Presented at the 111th Convention
2001 September 21–24 New York, NY, USA

An intelligent system approach to sound synthesis parameter optimisation

Brahim Hamadicharef and Emmanuel Ifeachor
Department of Communication and Electronic Engineering
University of Plymouth
Plymouth, Devon PL4 8AA, England
brahim@cis.plym.ac.uk ; E.Ifeachor@plymouth.ac.uk

ABSTRACT

An intelligent audio system for sound design using artificial intelligence techniques is reported. The system is used to analyse acoustic recordings, extract salient sound features and to process them to generate parameters for sound synthesis, in a manner that mimics human audio experts. Preliminary tests show that the use of the system reduces design time and yet the quality of the resulting sound is considered high by audio experts

INTRODUCTION

A new approach to sound design and modeling of musical instruments is presented. An intelligent audio system, based on fuzzy logic techniques, is used to analyse acoustic recordings, extract salient sound features and to process them to generate parameters for sound synthesis, mimicking human audio experts.

The main goal of our research is to investigate and develop artificial intelligence based techniques to capture and exploit audio expertise in the design of high quality sound. Our principal aim is to automate, as far as possible, the complex and time-consuming task of sound design for musical instruments, by exploiting the experience and knowledge of professionals such as musical instrument manufacturers, audio engineers and musicians.

The project is being carried out in collaboration with two audio companies, one of which has expertise in organ pipe sound synthesis. To our knowledge, this is the first attempt in computer music to capture and exploit, explicitly, knowledge from audio experts for sound design.

In this paper a description of the concept and implementation of an intelligent audio system is given together with preliminary results.

SOUND DESIGN ENVIRONMENT

The sound design environment depicted in Figure 1. It consists of an acoustic unit, an intelligent audio system and an electronic unit.

The acoustic unit is an organ pipe (other acoustic musical instruments may be used, but we are using organ pipe as a vehicle for the project). The sound of the organ pipe is recorded using microphones placed at different positions along the pipe. The intelligent audio system is a dedicated multimedia computer with a sound card and a MIDI a interface. The computer is also connected to the sound generator to transfer the setup parameters generated by the intelligent system. The electronic unit is based on a sound generator, supplied by our collaborative companies, connected to a MIDI master keyboard also controlling the multimedia computer via MIDI.

A digital mixer interconnects to all three units of the system. The audio expert or user can experiment within the sound design environment, e.g. play the acoustic instrument, record the sound onto disk, use the intelligent audio system to design sounds and listen to synthesized sound to assess their quality.

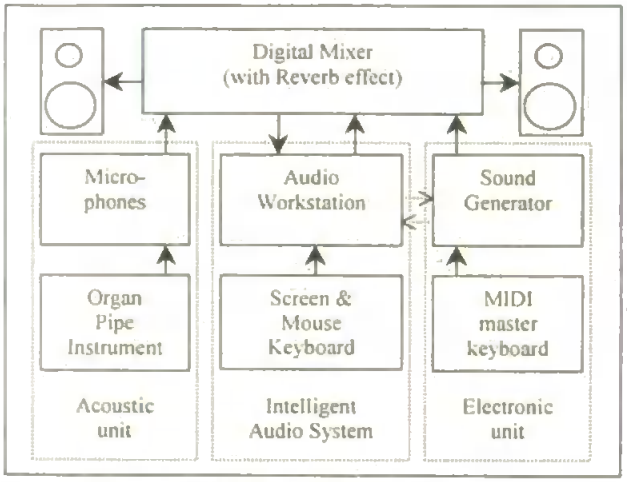


Figure 1: Diagram of the sound design environment.

The digital mixer may be used to add some reverberation to the synthetic sound to re-create a church acoustic environment typical to organ pipes.

INTELLIGENT AUDIO SYSTEM

A conceptual diagram of the intelligent audio system is shown in Figure 2. The intelligent audio system is implemented as software tools written within the MATLAB environment and linked to other research tools and custom applications written in C++. The intelligent audio system is divided into three main parts: a sound analysis engine, an intelligent system based on audio expertise and a sound synthesis engine.

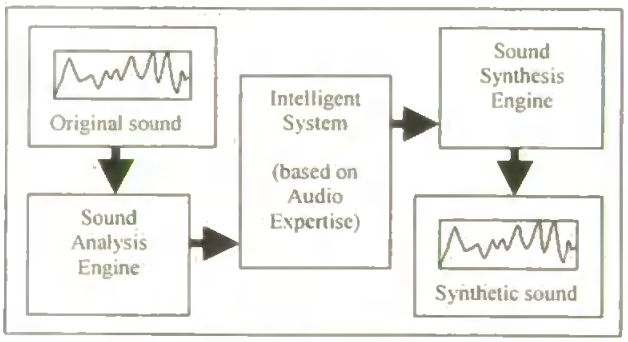


Fig. 2. Conceptual diagram of the intelligent audio system.

Sound analysis engine

The sound analysis engine serves as a front-end of the intelligent audio system. A block diagram of the sound analysis engine is shown in Figure 3. Using our sound design environment the audio expert or user can record organ pipe sounds and save them onto disk. In developing the intelligent audio system we have used a bank of sound from a CDROM. The sounds are standard WAV files, with two channels, the first being a recording very close to the mouth of the organ pipe and the other one near the extremity of the pipe. The user of the intelligent audio system loads a sound file and runs the sound analysis engine. For certain sounds we have found that some pre-processing is required to normalise the sound and remove background noise due to the recording conditions. Amplitude, frequency and phase trajectories are computed using phase vocoder-based techniques [1][2]. Examples of temporal and time–varying spectral representations of an organ pipe sound are given in Figure 4 and Figure 5 respectively.

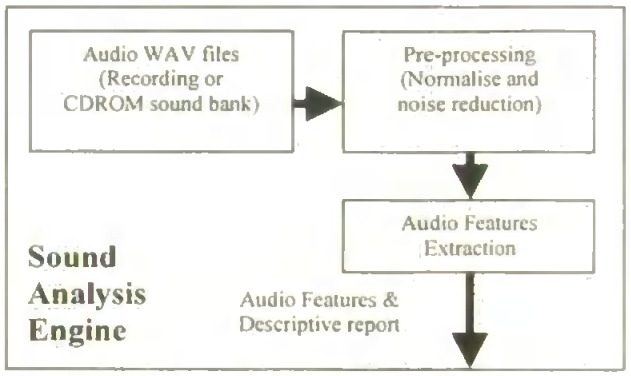


Figure 3: Block diagram of the sound analysis engine.

Audio features

The analysis engine further extracts audio features from time-varying harmonic components of the sound. The audio features can be divided into two categories: temporal and spectral features. Temporal features correspond to the time evolution of the overall amplitude envelope of the sound. The amplitude envelope has typically five segments: start, attack, sustain, release and end, with sometimes an additional decay segment between the attack and sustain. Spectral features related to time-varying evolution of the spectral envelopes.

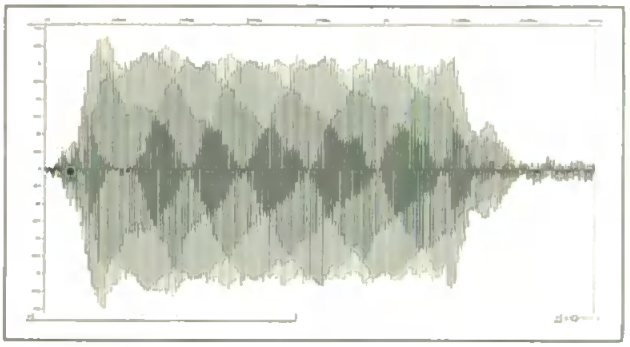


Figure 4: Temporal representation of an organ pipe sound.

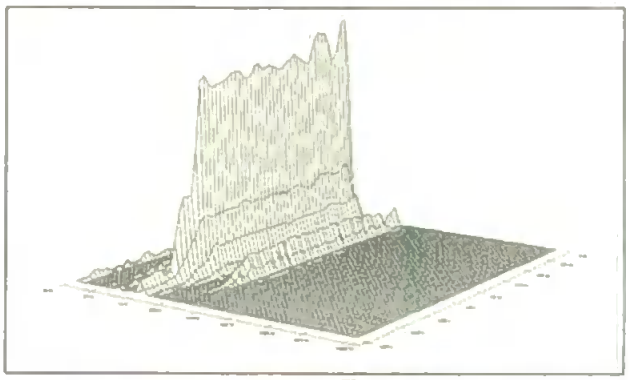


Figure 5: Time-varying spectral representation of an organ pipe sound.

Audio features extraction

In the following section methods used to extract the audio features will be detailed.

To extraction the temporal features of sound we model the amplitude envelope using a split-point time estimation technique [3]. This involves smoothing the envelope by convolution (digital signal processing technique) of the envelope with a gaussian. Using time derivatives of the smoothed envelope, the location of the start – end of each of the portion of the envelope can be determined (i.e. Attack, Decay, Sustain, Release). Each split-point has a variable amplitude (in percentage of the maximum amplitude of the partial) and time. The shape of each segment can be exponential, linear or logarithmic. This method has proved to be more stable than conventional methods.

The features extraction process focuses on 7 main spectral features: the energy of the fundamental, the energy of harmonics 2 to 4, the energy of the remaining harmonics, the frequency of the fundamental, the spectral centroid, the ratio of the energy of even harmonics to the total energy of the signal, and the ratio of the energy of odd harmonics to the total energy of the signal excluding fundamental [4][5].

Descriptive Report

A descriptive report is generated to characterise the general impression, the transient and steady state parts of the organ pipe sound. The linguistic terms are those used by audio experts to describe organ pipe sound. Similar work has recently been published in [6] (see table 2).

General impressions	old, noisy, pleasant, relaxed, simple, stable, strong, tensed, thin, undefined, unfocused, unpleasant, unstable, warm, weak.
Transient part	aggressive, strong, weak sounds like chiff, sounds like cough, sound like hiss, fast, gentle, long, short, slow, soft, connected, disconnected, integrated, related.
Steady state	airy, breathy, bright, clean, clear, cold, dirty, dull, floppy, flowy, fluffy, fluty, free, full, harsh, horn-like, leaky, loose, nasal, oppressive, reedy, rough, round, sandy, sharp, singing, splitting, stringy, thin.

Table 2: Linguistic descriptors used by the expert for the description of organ pipe.

Intelligent System

The intelligent system is based on fuzzy logic concepts [7][8] and is used to process audio features using rules provided by the audio experts to generate suitable parameters for the sound synthesis engine. A diagram of the fuzzy intelligent system engine is shown in Figure 6. First the fuzzification process transforms the extracted audio features into fuzzy variables. The fuzzy inference engine applies the rules to the fuzzy variables. The rules determine how the features are clustered / grouped based on the experience of audio experts. Finally, the output of the fuzzy inference engine are defuzzified and used to generate synthesis parameters to configure and control the sound synthesis engine.

Sound synthesis engine

The sound synthesis engine is currently based on a software tool that implements multiple wavetable synthesis with advanced modulation. It generates a sound file (WAV file) using the parameters from the intelligent system. A diagram of the sound synthesis engine is shown in Figure 7.

The audio expert can listen to the synthetic sound from the computer connected to the digital mixer and speakers (or headphones not shown in Figure 1). Listening and comparing the synthetic sound with the original allows the audio expert to assess the synthetic sound quality and hence the system performance.

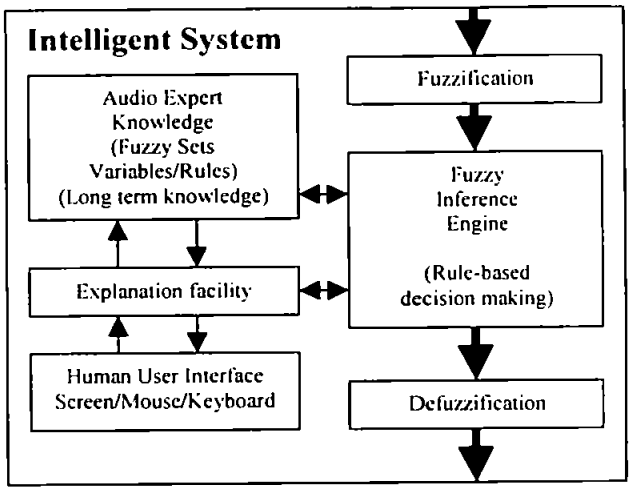


Figure 6: Diagram of the intelligent system.

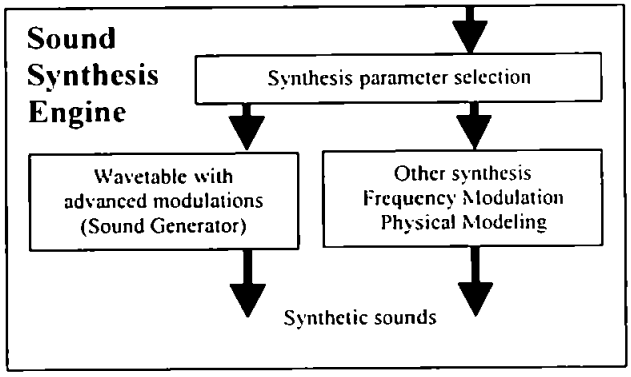


Figure 7: Diagram of the sound synthesis engine.

TESTS

To validate the new approach, preliminary tests have been conducted with a large bank of organ pipe sounds provided by one of the collaborative companies. It includes sounds from the Pedal (for the lower pitches that are played on the pedal board of the organ), Great, Swell and Choir departments of the organ pipe keyboard. Listening tests were conducted between the original sound and the sound generated from our intelligent audio system.

RESULTS AND PERFORMANCE

The preliminary results show that:

- The quality of the sounds generated by our system is often indistinguishable from the original one.
- The use of the system reduced the time taken to design organ pipe sounds to a quality that is considered high by two audio experts.

FUTURE WORK AND CONCLUSIONS

In future, the intelligent audio system will be extended in several ways. First we will to extend it to generate parameters for a complex, hardware-based, real time additive synthesis system which is of interest to the collaborative companies. The quality of the sound synthesis will be evaluated objectively using perceptual-based methods. At present, only subjective methods are used to assess the performance of synthesis systems of musical instruments. The rules used in the fuzzy expert system will to be refined to cater for sounds considered as more challenging by audio experts. Further, we will cater for other types of sound synthesis techniques such as Frequency Modulation [9][10] and

physical modeling (e.g. digital waveguides) [11][12]. Finally, we will investigate ways of extending our techniques to include other types of instruments such as bell [13] and piano [14].
An intelligent audio system for sound for organ pipes is reported. The intelligent audio system mimics human audio experts. To our knowledge, this is the first attempt in computer music to capture and exploit, explicitly, knowledge from audio experts for sound design.

ACKNOWLEDGEMENTS
The authors wish to acknowledge the support and assistance of Tony Koorlander and Graham Blyth of Musicom Ltd. We also would like to thank Rob Clark of Allen & Heath Ltd for his constructive suggestions and help.

References
[1] De Gotzen, A., Bernardini, N., and Arfib, D., 2000. "Traditional implementation of a phase vocoder: the tricks of the trade", Proceedings Workshop on Digital Audio Effects (DAFx-00), Verona, Italy,
[2] Laroche, J. and Dolson, M., 1999 "Improved phase vocoder time-scaled modification of audio". IEEE Transactions on Speech and Audio Processing. Vol. 7, No. 3 May, pp 323-332
[3] Jensen, F., 1999. "Envelope Model for isolated musical sounds", Proceedings Workshop on Digital Audio Effects (DAFx-99), Throndeim, Norway, 1999.
[4] Kostek, B., 1995. "Statistical versus artificial intelligence based processing of subjective test results" Proceedings of the 99th Audio Engineering Convention Preprint 4018 (P-3), Paris, France, February 25-28, 1995.
[5] Kostek, B., 1995. "Feature extraction methods for the intelligent processing of musical instruments" Proceedings of the 99th Audio Engineering Convention Preprint 4076 (H-4), New York, USA, October 6-9, 1995.
[6] Rioux and Vastfjall, 2001. "Analyses of verbal descriptions of the sound of the flue organ pipe". Musicae Sciencae, Volume 5, Number 1, Spring.
[7] Zahed, L. A., 1983. "The role of fuzzy logic in the management of uncertainty in expert systems", Fuzzy Sets and Systems, Vol. 11, pp. 199-227.
[8] Cox, E. The fuzzy system handbook, AP Professional
[9] Horner, A., 1998 "Nested Modulator and Feedback FM matching of Instrument Tones", IEEE Transactions on Speech and Audio Processing. Vol. 6, No. 4, July.
[10] Tan, B. T. G., and Lim, S. M., 1996. "Automated parameter optimisation for double frequency modulation synthesis using the genetic annealing algorithm", Journal of the audio Engineering Society, Vol. 44, No. 1/2, January/February 1996, pp. 3-15
[11] Smith, J. O., "Physical modeling using digital waveguides", Computer Music Journal, Vol. 16, No. 4, pp. 74-91, Winter 1992
[12] Valimaki "An improved digital waveguide model of a flute with fractional delay filters", 96th Nordic Acoustic Meetings, Helsinki, 12-14 June 1996
[13] Horner, A., Ayers, L. and Daniel Law, D. "Modeling Small Chinese and Tibetan Bells" Journal of the Audio Engineering Society, Vol. 45, No. 3, 1997 March
[14] Zheng, H., and Beauchamp, J., "Spectral characteristics and efficient critical-band-associated group synthesis of piano tones" Journal of the Acoustic Society of America, Vol. 106, No. 4, Pt. 2, pp. 2141-2142.

C.3 Paper 3 - AES115

Hamadicharef, B. and Ifeachor, E. C., "Objective Prediction of Sound Synthesis Quality" in Proceedings of the 115^{th} Audio Engineering Society convention, October 10-13, 2003 New York, USA.

Abstract

This paper is concerned with objective prediction of perceived audio quality for an intelligent audio system for modeling musical instruments. The study is part of a project to develop an automated tool for sound design. Objective prediction of subjective audio quality ratings by audio experts is an important part of the system. Sound quality is assessed using PEAQ (perceptual evaluation of audio quality) algorithm, and this greatly reduces the time-consuming efforts involved in listening tests. Tests carried out using a large database of pipe organ sounds, show that the method can be used to quantify the quality of synthesized sounds. This approach provides a basis for development of a new index for benchmarking sound synthesis techniques.

Audio Engineering Society

Convention Paper 5958

Presented at the 115th Convention
2003 October 10–13 New York, New York

Objective Prediction of Sound Synthesis Quality

Brahim Hamadicharef[1] and Emmanuel Ifeachor[1]

[1]Department of Communication and Electronic Engineering, University of Plymouth, Drake Circus, Plymouth Devon, PL4 8AA, UK
Correspondence should be addressed to Brahim Hamadicharef (bhamadicharef@plymouth.ac.uk)

ABSTRACT

This study is concerned with objective prediction of perceived audio quality for an intelligent audio system for modeling musical instruments. The study is part of a project to develop an automated tool for sound design. Objective prediction of subjective audio quality ratings by audio experts is an important part of the system. Sound quality is assessed using PEAQ (Perceptual Evaluation of Audio Quality) algorithm, and this greatly reduces the time-consuming efforts involved in listening tests. Tests carried out using a large database of pipe organ sounds, show that the method can be used to quantify the quality of synthesized sounds. This approach provides a basis for the development of a new index for benchmarking sound synthesis techniques.

1. INTRODUCTION

An important process in the development of sound synthesis systems, such as electronic musical instruments, is the assessment of the final perceived sound quality. Subjective listening tests with human subjects (audio experts) are commonly used to obtain accurate assessment of the final perceived sound quality. However, these tests are expensive, time consuming, require specialized sound facilities and need a large number of subjects to obtain the required accuracy. These problems have resulted in extensive research into objective audio quality metrics, i.e. computational methods that correlate well with human opinion. Increased knowledge and understanding of the complex human auditory system has recently resulted in objective quality metrics based on models of human perception [1]. Results have been promising, but much work still remains to be done before these metrics are widely adopted by the audio industry.

In this paper, objective prediction of perceived sound quality for an intelligent audio system is presented. This research project has been carried out in close collaboration with two audio companies, one of which has extensive expertise in sound synthesis of electronic pipe organs. To our knowledge, this is a unique attempt in computer music to capture and exploit, explicitly knowledge from audio experts for sound design and objective prediction of perceived sound quality.

The aim of this study is to undertake investigations into novel methods of assessing sound synthesis quality. This will form the basis of future developments of a new quality index to accurately and objectively predict sound quality for the benchmark of sound synthesis techniques.

The remainder of the paper is organized as follows. In Section 2, perceptual-based optimization of sound synthesis is briefly described. Section 3 describes the analysis of sound synthesis quality. Results are presented in Section 4 with pipe organ sound examples. Finally, Section 5 concludes this paper.

2. PERCEPTUAL-BASED OPTIMIZATION OF SOUND SYNTHESIS

The perceptual-based sound optimization of sound synthesis system, shown in Fig. 1, is part of the intelligent audio system described in [2]. It consists of four parts: a Sound Analysis Engine, a Knowledge-based Audio Feature Processing Engine, a Sound Synthesis Engine, and finally a Perceptual Error Analysis Engine. The system is used to automatically analyze acoustic recordings, extract salient sound features and process them to generate optimal sound synthesis parameters, mimicking human audio experts in the complex and time-consuming task of high quality sound design and modeling of musical instruments. We aim to fully automate the task of high quality sound design, based on the knowledge and experience of audio professionals, and to use an objective prediction of the subjective rating from listening tests by audio experts to assess the final perceived quality of sound synthesis techniques. The perceptual-based sound optimization of sound synthesis system has been implemented as a research software tool; a typical screenshot of the software tool is shown in Fig. 2.

The Sound Analysis Engine (block 1 in Fig. 1) is based on a phase vocoder analysis engine. It extracts the time varying evolution of the sound harmonic components both in amplitude and frequency. It also automatically estimates the Attack / Decay / Sustain / Release (ADSR) envelope segments. The Audio Feature Processing Engine (block 2 in Fig. 1) is our novel modeling method based on a fuzzy expert system developed in collaboration with two audio experts [2]. The fuzzy expert system emulates the decision making process of the human audio expert to generate a set of optimized sound synthesis parameters.

The Sound Synthesis Engine (block 3 in Fig. 1) is based on multiple wavetable sound synthesis with advanced modulation. The system generates sound files (standard wave files) that can be edited directly from the computer using monitor speakers or headphones, and also generates configuration files for our collaborator's electronic pipe organ musical hardware. The Perceptual Error Analysis Engine is the final part of the system (show as block 4 in Fig. 1) and is based on the PEAQ algorithm. It is used to control and optimize the knowledge-based audio features processing..

3. ANALYSIS OF SOUND SYNTHESIS QUALITY

Listening tests, which are the preferred way to assess perceived audio quality of sound synthesis, are known to be subjective, difficult to perform, time-consuming, expensive, and inconsistent. The ITU-R BS.1116 standard [4] gives guideline to perform these listening tests.

To predict the perceived quality of the sounds in an objective and reproducible manner the perceived sound quality engine exploits the Perceptual Evaluation of Audio Quality (PEAQ) algorithm [5] detailed in the ITU-R BS.1387 [6].

The original sound is used as the reference input signal and the test input signal is the synthetic sound.

PEAQ consists of a perceptual model, a feature extractor and a cognitive model. The perceptual model emulates the human hearing system, while the cognitive model reproduces the judgment made by human on the sound quality. Fig. 3 shows this generic perceptual measurement algorithm.

The outputs of the PEAQ algorithm, which includes model variables and measures of sound perception, are useful for characterizing sound synthesis artifacts as well as obtaining the final measure of sound quality (known as Objective Difference Grade (ODG) see Table 1). The Objective Difference Grade (ODG) is the output variable from the objective measure method, it ranges from 0 to –4, where 0 corresponds to imperceptible and –4 judged as "very annoying". In this work, the degradation corresponds to the difference in quality between the original sound (reference) and the synthetic sound (test) produced by the intelligent audio system.

In all our experiments, we have used the basic model of PEAQ algorithm a database of pipe organ sounds from our industrial collaborators. All the sound files are sampled at 48 kHz in 16-bit PCM. We have used the Opera "Voice/Audio Quality Analyser" from Opticom GmbH [7] and a modified version of EaQual [8]). The modified EaQual generates Matlab scripts to plot the 11 Model Output Variables (as shown in Table 2.) as well as the ODG and Distortion Index (DI).

4. RESULTS

Extensive tests were conducted using a large database of pipe organ sounds provided by one of our collaborative companies. The database included a

variety of pipe organ sounds recorded in churches and cathedrals across Europe and United States of America. Tests have been performed on individual pipe organ sounds, complete manuals (Great, Swell and Choir) and whole instruments. Final sound synthesis parameters were converted and loaded into an electronic pipe organ hardware provided by the collaborating company.

The perceived sound synthesis quality was assessed by audio experts rating during listening tests and then using Perceptual Error Analysis with the basic version of the PEAQ algorithm. The listening tests have been performed at our industrial collaborator facilities, which involved assessment of single note sound as well as extended piece of music.

To illustrate our work, we have chosen typical examples of pipe organ sounds from a sound database of a reference pipe organ in Texas, USA.

4.1. Choir sound

The first example is a sound from the choir manual. The sound analysis engine estimates the fundamental to be 108.352 Hz. This sound has very clean timbre. The sound analysis resulted in 222 harmonics. Some harmonics have similar time-varying evolutions and seem very dominants. The time domain waveform of this sound is shown Fig. 4. Fig. 5 shows the sound spectrum while Fig. 6 shows the harmonics analysis. In the harmonics analysis we can highlight the sustained part of each of the harmonics. This helps the audio expert who usually selects by hand the ADSR segments during the sound design process.

The sound synthesis analysis results are shown from Fig. 7 through Fig. 10. Fig. 7 illustrates the Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) averaged along the time axis versus the sound synthesis parameter (decreasing number of clusters). It shows that as the number of clusters is reduced (less resources used at the sound synthesis stage), the error increases. Fig. 8 shows a surface plot of the difference waveform error. Both give very poor real indication about the sound synthesis quality.

In Fig. 9, an ODG curve (averaged over all) showing minimal / mean / maximal values versus the number of clusters is shown. It indicates the performance of the synthesis technique and sound quality along the sound design process. As the number of clusters used in the synthesis decrease the sound synthesis quality decreases too. This curve could be broadly divided into three parts. The first part in which degradation increase slowly, then the degradation falls quickly and stabilizes towards the end.

The most interesting results are shown in Fig. 10. It presents an "ODG surface". The y-axis corresponds to the variation of a sound synthesis parameter (i.e. the number of clusters used in the synthesis) and x-axis is the time axis (frames). We have added a "perceptible threshold plan" (i.e. ODG equal to -1) that indicates the level at which PEAQ can detect that the degradation starts to be perceptible.

4.2. Flute Sound

The second example is a flute sound. The sound analysis engine estimates the fundamental to be 527.473 Hz. This sound has very clean timbre. The sound analysis resulted in 46 harmonics. Only one harmonic seems dominant and gives this sound a very sine wave like characteristic. The sound waveform is shown in Fig. 11, the sound spectrum in Fig. 12 and harmonics analysis in Fig. 13.

Results of the analysis of sound synthesis quality are show in Fig. 14 and Fig. 15. Reducing the number of clusters reduces the perceived sound quality. The average ODG (avgODG) curve presents a flat characteristic until it drops gently with a linearly characteristic. This indicates that the algorithm start reducing clusters with no perceptual degradation. The variation of between minimal / maximal values of ODG in Fig.14 is much more smaller than in previous example. Fig. 15 shows the ODG surface.

4.3. Principal Stop sound

The last example is a Principal Stop sound. The fundamental of this sound is 217.195.84 Hz. The sound analysis resulted in 111 harmonics.

The full analysis includes the sound waveform shown in Fig. 16, the sound spectrum in Fig. 17 and harmonics analysis in Fig. 18. Few harmonics have very stable behavior (compared to Choir sound example) giving this sound a defined characteristic.

Results of the analysis of sound synthesis quality are show in Fig. 19 and Fig. 20. Reducing the number of clusters reduces the perceived sound quality, in a more progressive way in this case.

Overall results show that only impairments between "imperceptible" and "perceptible, but not annoying" were considered acceptable by the audio experts. Visual inspection of the results in 3D (perceived audio quality versus synthesis parameter versus time) allows us to correlate the perceived audio quality with the sound attack, sustain, and release timing segmentation.

They also gives indications about the progression in the sound design process, and can be used to reveal imperfections of a sound synthesis technique and clustering rules in the case of our intelligent system. This is greatly helping to refine the rule-based fuzzy expert system and fine-tune the fuzzy sets and rules

of the fuzzy expert system used for as the Knowledge-based Audio Feature Processing Engine.

More audio examples and detailed results will be made available, before the convention, at the project home page [9]. .

5. CONCLUSIONS

In this paper, we have presented the objective prediction of sound synthesis quality looking at the final part of our intelligent audio system for perceptual-based optimization of sound synthesis. Audio experts' listening tests are now supported by a perceived sound quality assessment based on the ITU-R BS1387 Perceptual Evaluation of Audio Quality (PEAQ) algorithm.

Results from pipe organ sound database shows that plots of perceived sound quality versus number of clusters gives good indications about the progression in the modeling process, and can be used to reveal imperfections of a sound synthesis technique. It is used to improve clustering rules in the case of our intelligent system. This has proven to greatly help to refine the rule-based fuzzy expert system and fine-tune the fuzzy sets and rules of the fuzzy expert system engine.

Results also indicate that 6 to 12 clusters are often required for modeling the majority of pipe organ sounds, depending on the pitch of the sound and the type of pipe. This number can decrease to low as one or two for sounds with high pitch.

The system helps the audio experts to quantify the perceived sound quality of the synthesized sounds. It serves as a support tool, and helps to reduce time-consuming listening tests. However, more work is still needed to fully exploit the potential of objective measures of perceived audio quality in sound synthesis.

Future work should take into consideration the specificity of the PEAQ algorithm's cognitive model (audio coding artifacts as describes by Erne in [10]) and quality assessment of audio experts considered as having "Golden Ears" [11][12] when developing a new metric for sound quality index an expert level.

In future, we plan to investigate the use of knowledge gained using PEAQ as a basis for the development of a new index to accurately and objectively predict sound quality to benchmark sound synthesis techniques such as additive synthesis, wavetable synthesis, frequency modulation synthesis and digital waveguide modeling, with application to musical instruments such as piano [13] and church bells [14].

6. ACKNOWLEDGMENTS

This research is supported by the Engineering and Physical Science Research Council (EPSRC) under the grant (GR/S00859). We are grateful to Tony Koorlander and Graham Blyth for their help during the course of this project, and fruitful discussions. We also acknowledge support from the UK Government Science Research Investment Fund (SRIF) initiative for the purchase of the Opera "Voice/Audio Quality Analyzer" (Opticom GmbH).

7. REFERENCES

[1] E. Zwicker and H. Fastl, Psychoacoustics, Facts and models. Springer Verlag, 1999.

[2] B. Hamadicharef and E. C. Ifeachor, "An Intelligent System Approach To Sound Synthesis Parameter Optimisation," presented at the AES111th convention, New York, USA, 2001 November 30 – December 3.

[3] T. Koorlander, Personal email communications, 2002.

[4] ITU-R BS.1116,. "Methods for the subjective assessment of small impairments in audio systems including multichannel sound systems". 1994-1997

[5] T. Thiede, W. C. Treurniet, R. Bitto, C. Schmidmer, T. Sporer, J. G. Beerends, C. Colomes, M. Keyhl, G. Stoll, K. Brandenburg and B. Feiten , "PEAQ – The ITU standard for objective measurement of perceived audio quality," J. Audio Eng. Soc., vol. 48 pp. 3-29, 2000.

[6] ITU-R BS.1387, "Method For Objective Measurement of Perceived Audio Quality," 1998.

[7] Opticom, 2001. "OPERA: Voice/Audio Quality Analyser". Brochure and User Manual Version 3.5.

[8] A. Lerch, Personal email communications, 2002.

[9] http://www.tech.plymouth.ac.uk/spmc/S00859/

[10] M. Erne, "Perceptual Audio coders: What to Listen For," presented at the AES111th convention, New York, USA, 2001 November 30 – December 3.

[11] S. Shlien, and G. Soulodre, "Measuring the Characteristics of "Expert" Listeners," presented at the 101st Audio Eng. Soc., Los Angeles, USA, 1996, November 8-11.

[12] S. Shlien, "Auditory Models for Gifted Listeners," in J. Audio Eng. Soc., vol. 48, pp. 1032-1044, November 2000.

[13] J. Laroche, and J. L. Meiller, "Multichannel Excitation/Filter modeling of percussive sounds with application to the Piano," in IEEE Transactions on Speech and Audio Processing, vol. 2, pp. 329-344, IEEE, April 1994.

[14] M. Karjalainen, V. Välimäki, and P. A. A. Esquef, "Efficient Modeling and Synthesis of Bell-like Sounds," in Proc. of the 2002 Conference on Digital Audio Effects, pp. 181-186, DAFX, September 2002.

ODG value	Meaning
0.0	Imperceptible
-1.0	Perceptible but not annoying
-2.0	Slightly annoying
-3.0	Annoying
-4.0	Very annoying

Table 1: Objective Difference Grade with meaning

Model Output Variables	Purpose
WinModDiff1	Changes in modulation
AvgModDiff1	(related to roughness)
AvgModDiff2	
RmsNoiseLoud	Loudness of the distortion
BandWidthRef	Linear distortions
BandWidthTest	(frequency response, etc.)
RelDisFrame	Frequency of audible distortions
Total NMR	Signal-to-mask ratio
MFPD	Detection probability
ADB	Detection probability
EHS	Harmonic structure of the error

Table 2: Model Output Variables (MOVs)

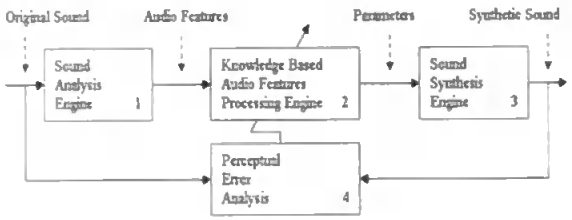


Fig. 1: Perceptual-based optimization of sound synthesis

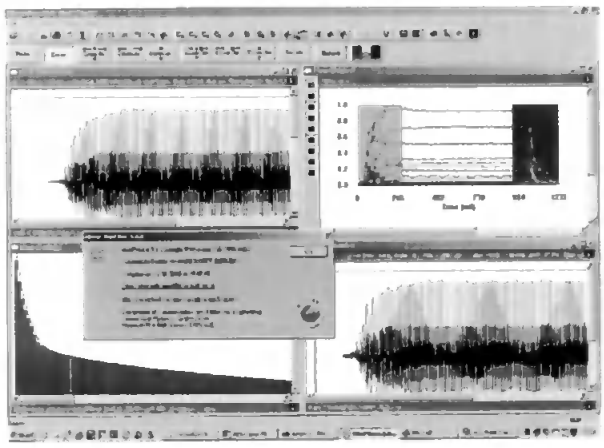


Fig. 2: Screenshot of the software tool

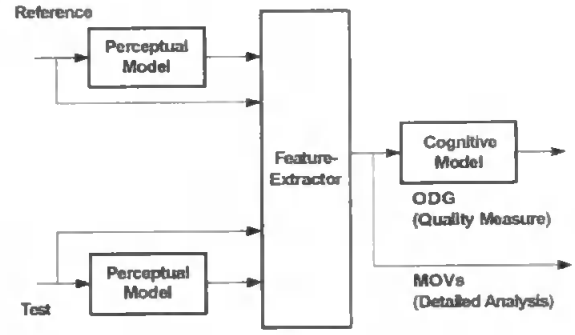


Fig. 3: Generic perceptual measurement algorithm

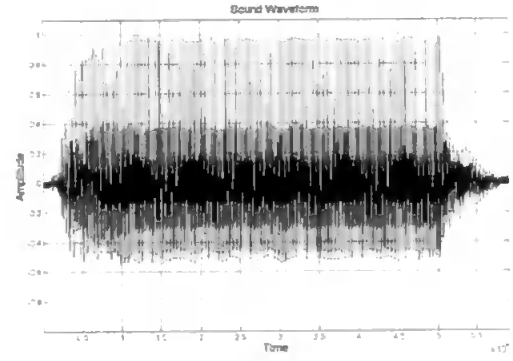


Fig. 4: Choir waveform sound analysis

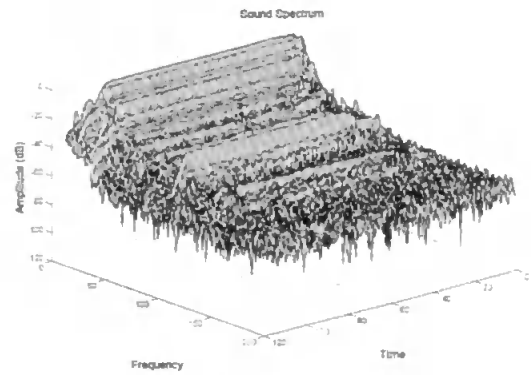


Fig. 5: Choir sound spectrum analysis

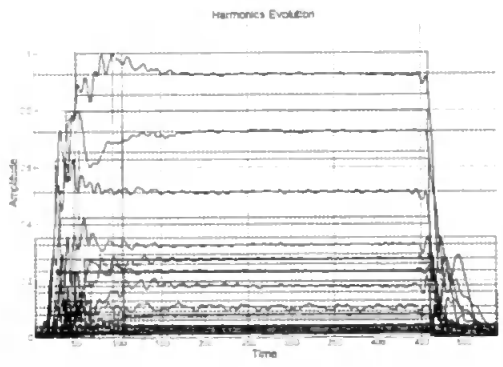


Fig. 6: Choir sound harmonic analysis

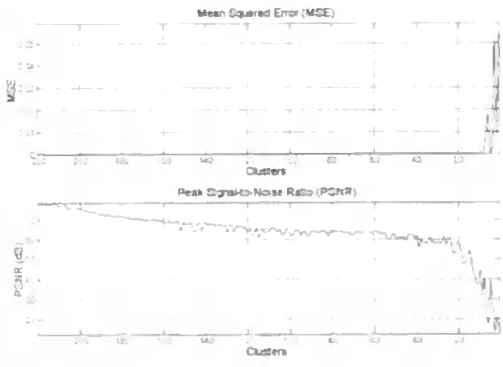


Fig. 7: Choir sound synthesis MSE / PSNR error

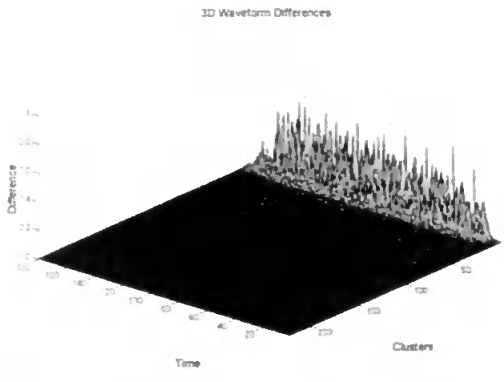


Fig. 8: Choir waveform difference error

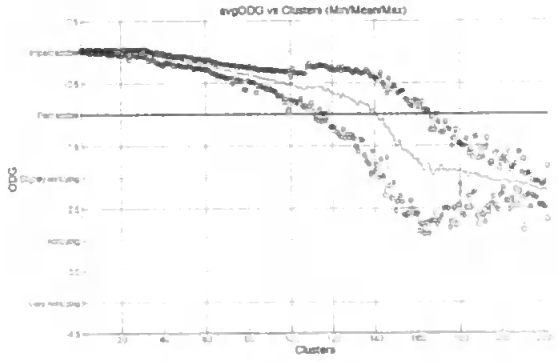


Fig. 9: Choir ODG versus clusters

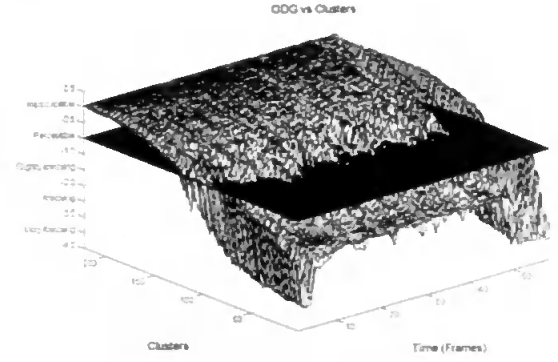


Fig. 10: Choir ODG surface

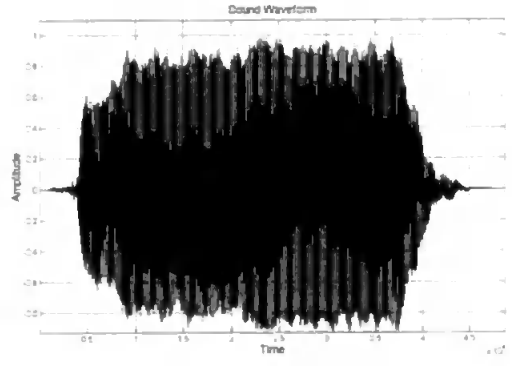


Fig. 11: Flute sound waveform analysis

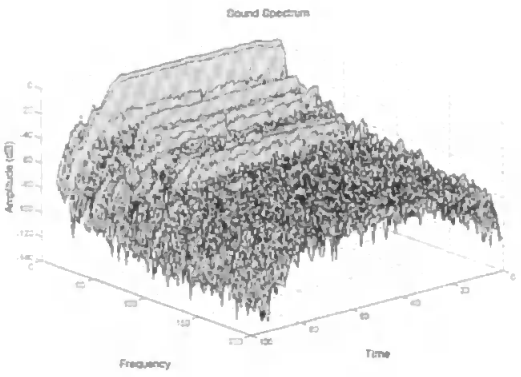


Fig. 12: Flute sound spectrum analysis

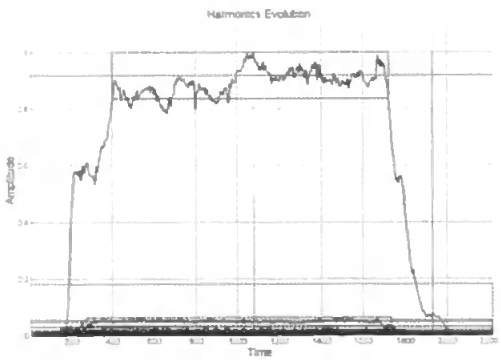


Fig. 13: Flute sound harmonic analysis

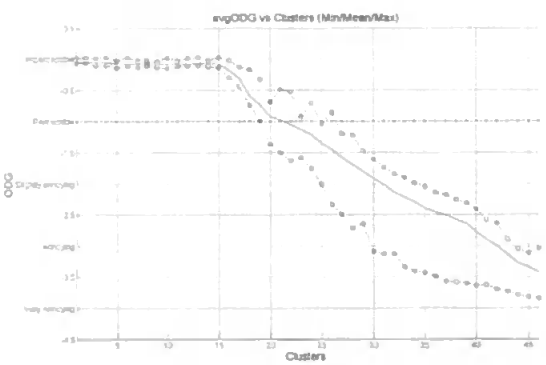


Fig. 14: Flute ODG curve

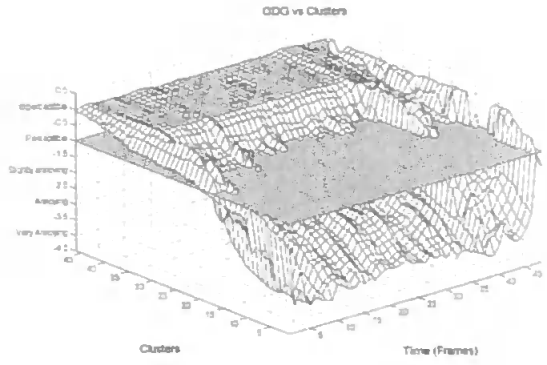


Fig. 15: Flute ODG surface

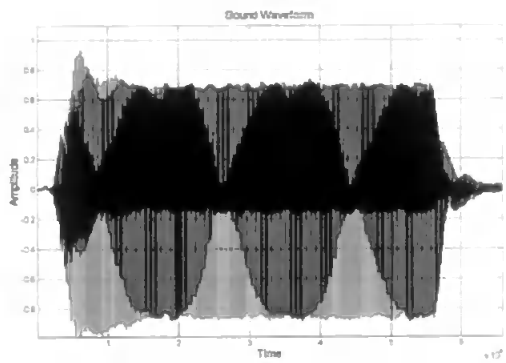


Fig. 16: Principal Stop sound waveform analysis

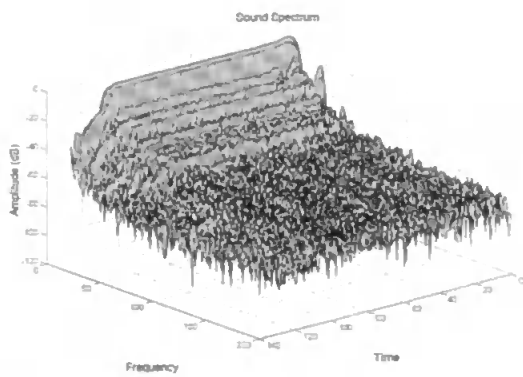


Fig. 17: Principal Stop sound spectrum analysis

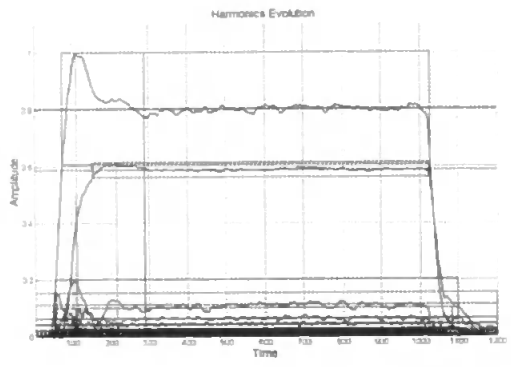


Fig. 18: Principal Stop sound harmonic analysis

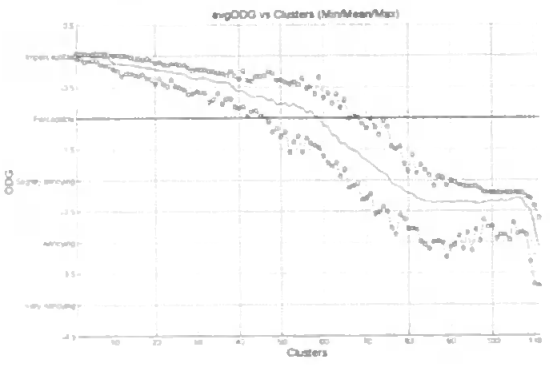


Fig. 19: Principal Stop ODG curve

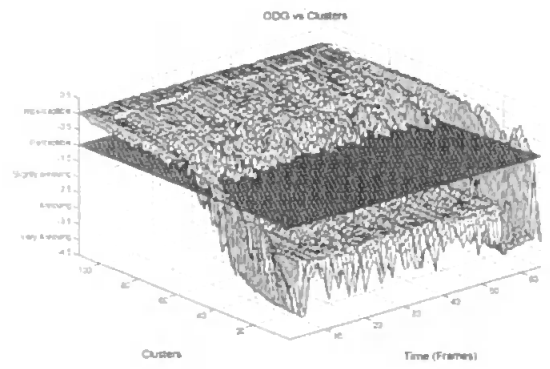


Fig. 20: Principal Stop ODG surface