TARGET CLASSIFICATION AND RECOGNITION USING UNDERWATER ACOUSTIC SIGNALS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

TAYFUN YAĞCI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
THE DEPARTMENT OF COMPUTER ENGINEERING

JULY 2005

Approval of the Graduate School of Natural Applied Sciences

Prof. Dr. Canan ÖZGEN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Ayşe KİPER
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Ahmet COŞAR
Supervisor

Examining Committee Members

Prof. Dr. Faruk POLAT	(METU, CENG)
Assoc. Prof. Dr. Ahmet COŞAR	(METU, CENG)
Assoc. Prof. Dr. İ. Hakkı TOROSLU	(METU, CENG)
Asst. Prof. Dr. Halit OĞUZTÜZÜN	(METU, CENG)
Assoc. Prof. Dr. İlyas ÇİÇEKLİ	(Bilkent Unv.)

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work

Name. Last Name : Tayfun YAĞCI

Signature :

ABSTRACT

TARGET CLASSIFICATION AND RECOGNITION USING UNDERWATER ACOUSTIC SIGNALS

YAĞCI, Tayfun
M. S., Department of Computer Engineering
Supervisor: Assoc. Prof. Dr. Ahmet COŞAR

July 2005, 116 pages

Nowadays, fulfillment of the tactical operations in secrecy has great importance for especially subsurface and surface warfare platforms as a result of improvements in weapon technologies. Spreading out of the tactical operations to the larger areas has made discrimination of targets unavoidable. Due to enlargement of the weapon ranges and increasing subtle hostile threats as a result of improving technology, "v*isual*" target detection methods left the stage to the computerized acoustic signature detection and evaluation methods.

Despite this, the research projects have not sufficiently addressed in the field of acoustic signature evaluation. This thesis work mainly investigates classification and recognition techniques with TRN / LOFAR signals, which are emitted from surface and subsurface platforms and proposes possible adaptations of existing methods that may give better results if they are used with these signals. Also a detailed comparison has been made about the experimental results with underwater acoustic signals.

Keywords: Target Radiated Noise, Low Frequency Analysis and Recording, Classification, Pattern Matching, Underwater Acoustic Physics.

ÖZ

SUALTI AKUSTİK SİNYALLERİNİ KULLANARAK HEDEF SINIFLANDIRMA VE TANIMA

YAĞCI, Tayfun
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Danışman: Doç. Dr. Ahmet COŞAR

Temmuz 2005, 116 sayfa

Günümüzde, silah teknolojisindeki gelişmenin sonucu olarak su üstü ve özellikle de sualtı platformlarının harekat görevlerini gizlilik içinde icra etmeleri büyük önem kazanmıştır. Harekatın daha geniş alanlara yayılması da hedef tespitine ek olarak hedef ayrımının yapılmasını zorunlu hale getirmiştir. Gelişen teknoloji ile birlikte silah menzillerinin ve gizli düşman tehdidinin artması nedeniyle görsel hedef tespit metotları sahneyi bilgisayarlı akustik parmak izi tespit ve değerlendirme metotlarına bırakmıştır.

Buna rağmen akustik imza değerlendirmesi alanındaki çalışmalar henüz yeterli düzeyde değildir. Bu tez temel olarak sualtı ve su üstü platformlarından yayılan TRN/LOFAR sinyallerini sınıflandırma ve tanıma tekniklerini incelemekte, ve bu sinyallerle beraber kullanıldığında daha iyi sonuçlar verebilecek mevcut metotlara uygulanabilecek uyarlamalar öne sürmektedir. Ayrıca sualtı akustik sinyallerinin kullanıldığı deneysel sonuçlar hakkında detaylı bir değerlendirme ve karşılaştırma yapılmıştır.

Anahtar Kelimeler: Hedeften Yayılan Gürültü, Alçak Frekans Analiz ve Kayıt, Sınıflandırma, Örüntü Eşleme, Sualtı Seda Fiziği.

To My Parents

ACKNOWLEDGEMENTS

I would like to thank my advisor Assoc. Prof. Ahmet COŞAR for his guidance and support in all stages of this work. I am most grateful and also give my special thanks to my wife Canan YAĞCI for her support and patience throughout this study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AGC	Automatic Gain Control
ANN	Artificial Neural Network
BPNN	BackPropagation Neural Network
BR	Blade Rate
CWT	Continuous Wavelet Transform
DEMON	Detection of Envelope Modulation of Noise
DFT	Discrete Fourier Transform
DTW	Dynamic Time Warping
DWT	Discrete Wavelet Transform
EDF	Euclidian Distance Function
EM	Expectation and Maximization
FFT	Fast Fourier Transform
FIR	Finite Impulse Response filters
FT	Fourier Transform
GLA	Generalized Lloyd Algorithm
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
IDFT	Inverse Discrete Fourier Transform
IIR	Infinite Impulse Response filters
LBG	Linde Buzo Gray
LDA	Linear Discriminant Analysis
LDB	Local Discriminant Basis
LOFAR	Low Frequency Analysis and Recording
LP	Linear Prediction
LPCC	Linear Predictive Cepstral Coefficients
LTI	Linear Time Invariant
LVQ	Learning Vector Quantization
MFCC	Mels Frequency Cepstral Coefficients

MPN	:...........................McCulloch-Pitts Neuron
MRA	:...........................Multi Resolution Analysis
MSE	:...........................Mean Square Error
NPM	:...........................Normal Perona Malik
PC	:...........................Principal Component
PCA	:...........................Pricipal Component Analysis
PDF	:...........................Partial Differential Equation
PLP	:...........................Perceptual Linear Prediction
PNN	:...........................Pairwise Nearest Neighbor
PSR	:...........................Propeller Shaft Rate
QMF	:...........................Quadrature Mirror Filter
RLS	:...........................Randomized Local Search
RPM	:...........................Regularized Perona Malik
SOM	:...........................Self Organizing Map
STFT	:...........................Short Term Fourier Transform
TRN	:...........................Target Radiated Noise
UMF	:...........................Uniform Modulated Filterbank
UWAC	:...........................UnderWater Acoustic Channel
WPD	:...........................Wavelet Packet Decomposition
WTA	:...........................Winner Takes All

CHAPTER 1

INTRODUCTION

A few decades ago, vision was the primary target detection method for conduction of tactical operations. Nowadays, that driving method, vision, left the stage to the acoustic signature and electromagnetic trace detection due to the effect of improving technology on the weapon ranges and increment of the subtle hostile threats. Almost every naval forces of the world have high range weapons in their inventory, which can be launched beyond the visual range. So that makes the naval forces having the ability of detecting targets or classification of them by using different methods especially acoustic signature evaluation indispensable.

This research investigates the concept of underwater acoustic signature evaluation rather than electromagnetic trace detection. Improvements of the sonar technology after World War II has led us make Low Frequency Analysis and Recording easily. Today most of the sonar equipments used by submarines and Anti Submarine Warfare platforms have the ability of omni directional surveillance, high capacity of recording, detailed digital signal processing capable processors, DEMON analysis modules etc.

In spite of the SONAR technology capabilities and the increment of subtle hostile threat that, comparatively the research projects are not adequate in the field of classification methods of TRN / LOFAR signals that are utilizable with acoustic signature evaluation.

Detection and the classification of surface and subsurface platforms from underwater acoustic signals expose several technical difficulties if compared with speech processing. Also several factors contribute to make detection, classification and discrimination processes very hard to deal with. These factors are, non-repeatability and variation of the target signature for range and type of sonar, low frequency backscattered clutter caused by biological resources, environmental effects on acoustic signal like salinity, density, inhomogeneous seawater, and temperature effects on speed, reflection and refraction of sound, surface and bottom reverberation, non-stationary acoustic signatures of the target which depends on the speed, course and operational behaviors. Despite that, a new robust classification and recognition schema is needed to overcome all the factors we faced above.

In legacy speech processing methods, the process is mainly centered around three phases. Feature extraction from digitally recorded signal and dimensionality reduction, classification of the signal represented by extracted features and if needed recognition of the signal. Last two processes can be thought as one process since the same methods are used. There is a decision making phase additionally in the latter one. Classifiers can tolerate the variations in the feature space and in some cases this leads classifiers no longer be able to capture the temporal and spectral properties of the varying signals. This causes misclassifications also. In addition, new threats and non-targets that were not in the original training data sets may enter the field of view of the sensor. In junction with the other side effects of underwater acoustics, we face with a difficult and challenging signal-processing problem [28]. Due to the weakness of classification system, the dominant process, which discriminates signature of the classes, becomes Feature Extraction.

Various signal processing schemes and statistical approaches applied for feature extraction and some of them were tested for three main classification processes. Back Propagation Neural Networks, Gaussian Mixture Models and Kohonen Self Organizing Maps.

The aim of this thesis is not to choose the best processes in the literature by means of classification and discrimination of targets automatically. This research project represents comprehensive evaluation and comparison on combinations of existing and modified processes.

1.1. Problem Statement & Research Question

This thesis tries to find the answer of research question " *How can I develop a computer system that gives the Naval Ships and Submarines the ability of classification and discrimination of surface or subsurface targets using TRN & LOFAR signals.*" Considering the research question above, more formally, we can define the problem statement as follows.

There are many sound components, which should be handled by sonar equipments in the environmental domain of underwater acoustics. It is important to *pick the relevant low frequency sound components*, which can be used as a signature of targets. Naval ships and submarines can benefit from research that *identifies the target signature* from that scattered signal environment for surveillance and engagement tasks.

Nowadays most of the target classification systems in Navy depend on the operator's visual and hearing skills. If we consider that education of a high-qualified sonar operator takes at least ten years on duty, we cannot neglect the computer system aid, which makes the classification automatically.

Also another approach comes to the stage, if operators can make the classification of targets manually, is it possible to use existing speech recognition techniques in the field of classification problem for underwater acoustic signatures? Also high-qualified signal processing capabilities of sonar systems help us in processing acoustic signals over human hearing system range.

1.2. Aim

Aim of this research project is to develop an efficient system that can classify and recognize TRN & LOFAR signals, emitted from surface or subsurface targets and recorded by the passive sensors, automatically by using existing speech or non-speech classification methods and to show that existing methods are useful if we apply some modifications in order to exempt environmental specific restrictions of underwater acoustics discussed in this study.

General Acoustic Signal processing and classification schema, in which the candidate schema we want to produce will take place can be drawn as follows.

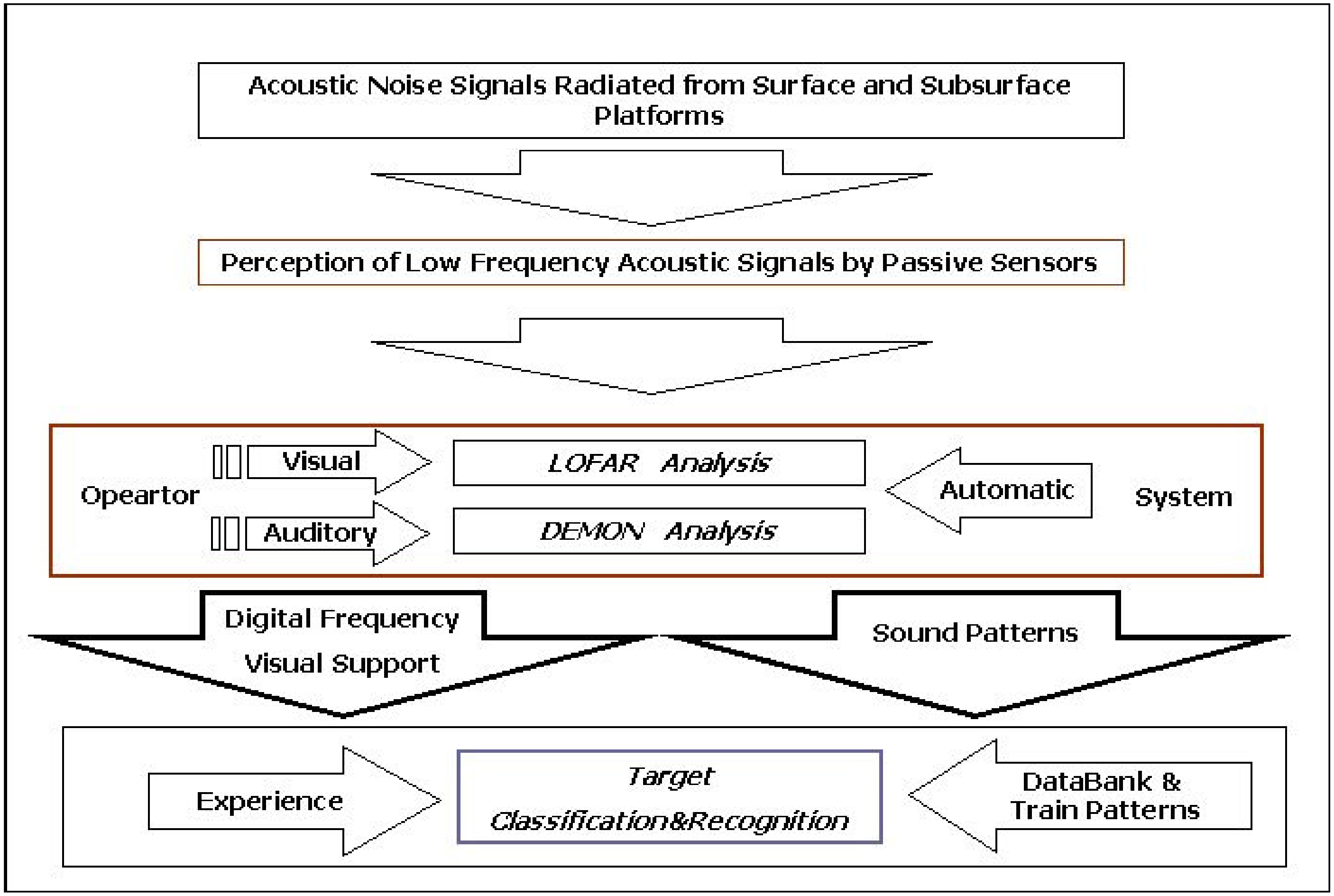

Acoustic Noise Signals Radiated from Surface and Subsurface Platforms
Perception of Low Frequency Acoustic Signals by Passive Sensors
Opeartor
Visual
Auditory
LOFAR Analysis
DEMON Analysis
Automatic
System
Digital Frequency
Visual Support
Sound Patterns
Experience
Target
Classification&Recognition
DataBank &
Train Patterns

Figure 1: General acoustic signal processing and classification scheme.

1.3. Hypothesis

This research project premises that detection of valuable parameters from an acoustic signal emitted from target sources and matching it to a known pattern is a classification and recognition problem. Underwater acoustic domain consists of large spectrum of sounds so; the acoustic signals, which can be considered as a part of acoustic signature of a target platform, constitute just a small subset of this spectrum. As a result, we have to define two main methodologies. First discriminate this valuable subset from the vast area of underwater acoustic environment and process it to form an acoustic fingerprint, which obeys some standards. The latter methodology should take the fingerprint as an input and use those parameters to improve basic class library. Also we should define a systematic way to apply data mining or pattern recognition techniques (or both) on those parameters and to make class library for target signature identification.

1.4. Organization of This Document

Remainder of this research project was organized in five chapters as following.

Chapter 2: In this chapter, it was aimed to provide basic knowledge of the underwater acoustic physics. Basic information about the specific properties of underwater acoustic was given and some clues were revealed about how sonar operators do their jobs. Also preliminary digital signal processing techniques, which will be used in acoustic signal processing methods, were described. Basic information about those processing methods and application strategies were investigated.

Chapter 3: Feature extraction methodologies were discussed. The first main methodology we have mentioned as hypothesis and using the knowledge we had from preceding chapters were covered.

Chapter 4: This chapter covers the second main methodology of hypothesis. Legacy pattern recognition and data mining techniques were discussed and probable modifications were proposed.

Chapter 5: This chapter shows the experimental results and draws comparisons of all methodologies we have used along the research progression.

Chapter 6: In this section, conclusions about the thesis framework and our experimental results were presented and the answers of the problem statement were evaluated. Some clues about possible future extensions, which were not studied in this work, of complete structure of acoustic signature detection were presented and future research methodologies motivated by this research project were discussed.

CHAPTER 2

BACKGROUND INFORMATION

In this chapter, background information about the “Underwater Acoustic Physics” and “Digital Signal Processing” has been surveyed.

2.1. Underwater Acoustic Physics

As we mentioned before, we will use TRN & LOFAR signals as input variables. Those signals are propagated in underwater and recorded by passive sensors. So we have to know the behaviors of sound in water, which is exactly not the same as in air.

2.1.1. Factors That Effect Sound Velocity

Sound velocity in seawater is a function of temperature, salinity and the pressure of the water and those parameters change with the depth, season, geographical location, and time. Generally changes observed in temperature, depth and salinity change the sound velocity accordingly. Sound velocity depends on the carrier medium and is 4800 ft/s in the seawater with the salinity 0,035 and the temperature 39 °F. The driving factor that influences the sound velocity is the temperature. 1 °F of increment in temperature increases the sound velocity 6 ft/s. When it goes deep, the pressure increases and also the sound velocity increases as well so the sound refracts to the low-pressure side.

100 ft increment in depth increase the sound velocity 2 ft/s, and also 0,1 percent increment in the salinity increases the sound velocity 4,5 ft/s as well [3,6,12,49]. Sound refracts from high-density part of the water to low-density area. Sound has lazy characteristic, hence it propagates to the place where it has lowest velocity. All the effects of temperature, depth, and the salinity were graphically illustrated in *Figure 2*.

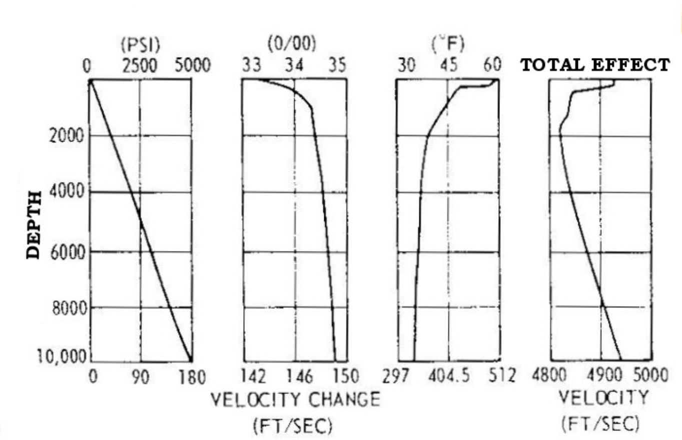


Figure 2: Depth, Salinity and Temperature effects on sound velocity.

2.1.2. Fundamental Acoustic Energy Losses

There are some losses that occur in sound energy during the propagation of sound waves in seawater. Those losses, basically, makes acoustic characteristic of radiated noises different from speech signals. They can be grouped as

- Propagation Loss
- Absorption of Sound
- Reverberation
- Scattering

2.1.3. Underwater Acoustic Signal Processing

In this section, superficial discussion about Underwater Acoustic Signal Processing techniques will be made and some basic information about LOFAR signal analysis will be presented.

Acoustic signals emitted from target platforms can be evaluated for different purposes. If we draw the general schema about acoustic signal evaluation, it is absolute that we gather a lot of useful information required by tactical operations.

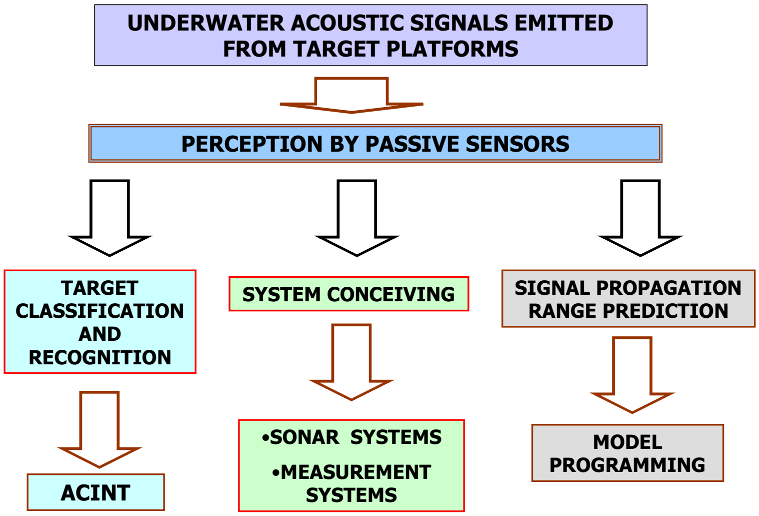

UNDERWATER ACOUSTIC SIGNALS EMITTED FROM TARGET PLATFORMS
PERCEPTION BY PASSIVE SENSORS
TARGET CLASSIFICATION AND RECOGNITION
SYSTEM CONCEIVING
SIGNAL PROPAGATION RANGE PREDICTION
ACINT
•SONAR SYSTEMS
•MEASUREMENT SYSTEMS
MODEL PROGRAMMING

Figure 3: Acoustic signal evaluation diagram.

Acoustic signals radiated from surface or subsurface platforms propagate through the seawater and during this diffusion, some information losses and signal metamorphosis happens as a result of underwater acoustic physics mentioned earlier.

Those acoustic signals commonly consist of wave signals that target platforms behaviors bring about. Basically these signals grouped into two main categories. “Broadband and Narrowband signals” [6] Below this two main category, we classify the signals caused by the targets machinery into sub-groups. The aim of this thesis is not exactly making the classification according to the taxonomy shown in *Figure 4*, since there is high correlation between the classes in that fashion. For example one submarine can make cavitations and emits auxiliary machines noise together although sometimes it can be so quiet that you cannot hear from 100 yards, or both a frigate and freighter can have the same auxiliary machine with the same characteristics working at the same time. This taxonomy problem may be solved but it was left for future researches.

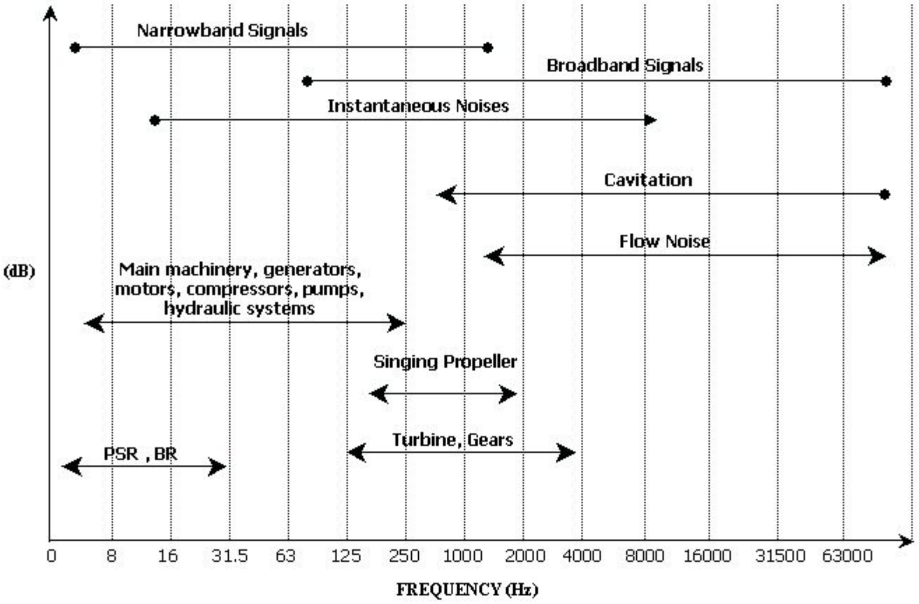


Figure 4: Frequency range diagram of specific noise groups

In order to make a fulfillment of a successful sound analysis, capabilities of the receiving component is as influential as the characteristics of the radiated signal. The receiving component is commonly “ *SONAR* ” equipments, which has the capability of LOFAR and DEMON analysis. Some artifacts may emerge during the signal-processing phase since sonar equipments are highly complicated microelectronic-computer systems. The term “*Artifact*” used here is the noise resulted from interior structure of the equipment, not a result of target noise [6].

The acoustic signals processed by sonar equipment are recorded, after some amplification and filter process, and stored for classification purposes. Automatic analysis and classification by equipment are not like usual recognition and classification processes. Sonar acquires raw sound data from hydrophones, applies signal-processing applications, and presents “ *LOFARGRAM* ” data, which is almost the same as the waterfall diagram of time frequency spectrum. There are measurements of some statistical information help the operator to make analysis, as well.

As the purpose of this research, we are interested in the automatic classification of underwater signals emitted from target sources.

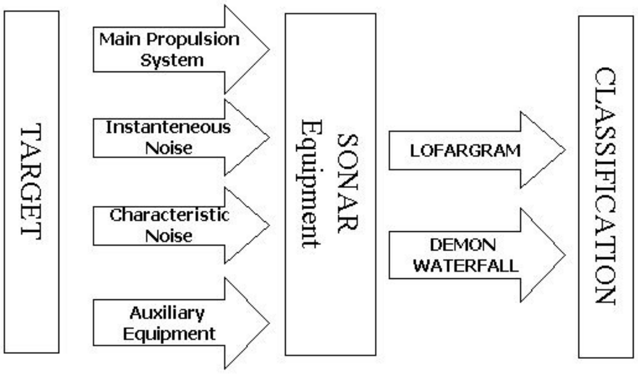


Figure 5: General signal classification process

2.2. Digital Signal Processing

This section describes important digital signal processing methods used in LOFAR analysis, skeleton of the Classification – Recognition system and Feature Extraction.

2.2.1. Analog / Digital Conversion

Similar to the speech signal, which is a form of wave motion carried by a medium; a microphone like equipment "*Hydrophone*" that consists of piezoelectric ceramic materials can capture underwater acoustic sound waves. Hydrophones convert the continuous water pressure changes into voltage changes. The analog signal $X_a(t)$ is then sampled to a digital form X (n) by an *analog-to-digital converter*. The A/D Converter samples the analog signal uniformly with the *sampling period* T [23].

$$X[n] = X_a(nT) \qquad 2.1$$

The inverse of T is the sampling frequency and marked as $F_S = 1 / T$. Given that the original signal $X_a(t)$ contains frequencies up to $F_S / 2$ that is called *Nyquist rate* and it is the upper limit for frequencies present in the digital signal.

Nyquist proved that, for a noiseless channel, if an arbitrary signal has been run through a low-pass filter of bandwidth H, the filtered signal could be completely reconstructed by making exactly 2H samples per second [45].

$$\text{Maximum data rate } = 2H \log_2 V \text{ bits/sec} \qquad 2.2$$

In order to recover the signal X_a (t) from it's samples exactly, it is necessary to sample X_a (t) at a rate greater than twice it's highest frequency component[1].

[1] Sampling Theorem and the proof can be found online at http://cnx.rice.edu/content/m11443/latest/, Anders Gjendemsjo's web page.

According to the theorem above, in order to preserve frequencies up to 4 kHz, the sampling rate, F_s, must be chosen more than 8 kHz. A/D converter also *quantizes* the samples into finite precision. The number of bits used per sample determines the dynamic range of the signal. Adding one bit extends the dynamic range of the signal roughly +6 dB[2].

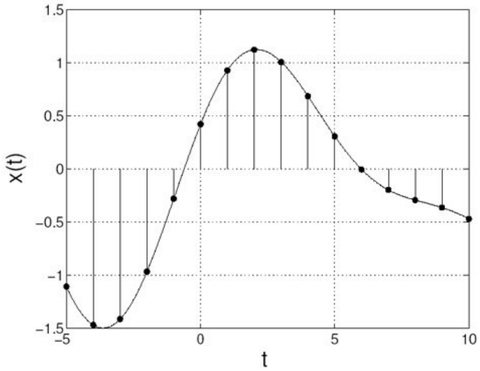


Figure 6: Analog signal sampled with dots[3]

The analog signal can be reconstructed from the sampled form by the following formula.

Note that at the original sample instances " t = nT ", the reconstructed analog signal is equal to the value of the original analog signal [34].

[2] Ifeachor E., Lewis B., "Digital signal processing – a practical approach, 2nd ed.", Pearson Education Limited, Edinburgh Gate, 2002, cited in [24].

[3] Figure has been taken from Anders Gjendemsjo's web page. This page is available at http://cnx.rice.edu/content/m11443/latest/

2.2.2. Fourier Analysis

Fourier analysis is a useful tool that decomposes a signal into the spectral properties (complex exponential functions) of different frequencies. A sound signal constitutes from different sinusoidal basis functions of different frequencies, amplitudes, and phases. Fourier analysis provides a way of extracting different underlying frequency components of the signal, or synthesizing the original time domain signal from the frequency domain representation of it [23].

2.2.2.1. Fast Fourier Transform

To approximate a function by samples, and to approximate the Fourier integral by the discrete Fourier transform, requires applying a matrix whose order is the number sample points n. Since multiplying a n x n matrix by a vector has the computational complexity of $O(n^2)$, the problem gets quickly worse as the number of sample points increases [17].

However, if the samples are uniformly spaced, then the Fourier matrix can be factored into a product of just a few sparse matrices, and the resulting factors can be applied to a vector in a total of order O(n log n) computational complexity [17,23].

Input signal vector must have length of power of two, $2^m, m \in \mathbb{N}_+$, as the main requirement of fast Fourier transform. If the signal length is below this range, the signal is zero-padded and after that FFT is applied [23]. Zeros can be added to the beginning or end of the signal and this process does not affect the result [36].

If we talk about the computation timesaving in practice, for 1024 samples, the ratios of DFT to FFT are 200 for multiplications and 100 for additions.

2.2.2.2. Short Term Fourier Transform

For a stationary signal, all the transforms we mentioned above are concluded with thriving results. We use the term *stationary* for the signals, which have the fixed frequency components from the beginning to the end. If the signal consists of some frequency components for each of them starts or ends at different t time (*non-stationary*), we face a problem with classical Fourier transforms.

The information provided by the integral in transform formula corresponds to all time instances, since the integration is from minus infinity to plus infinity over time. It follows that no matter where in time the component with frequency f appears, it will affect the result of the integration equally as well. No matter the frequency component f appears at different times t_x, it will have the same effect on the integration. That means Fourier transform lose time information if the signal is non-stationary, then the result obtained by the Fourier transform makes no sense, also time information contains useful clues about the characteristics of the signal [35].

The Fourier transforms indicates just the signal contains the frequency component f at any time. This restriction was left to the section Wavelet Transforms on the point of a comprehensive discussion.

Researches finished their studies by applying some minor modifications to the FT. STFT can be described as the following equation [39].

$$STFT_x^{(\omega)}(t,f)=\int_t \left[x(t)\omega^*(t-t') \right] e^{-2j\pi ft}dt \qquad 2.3$$

The solution was found by multiplication of the signal with a *window function*. The window function has the same length with the segment of the signal, which is assumed to be stationary applied after the signal was segmented into small enough particles that meet our needing in the sense of stability [23,39].

A windowed DFT is computed by starting at t_0, the window applied to the signal for every time segment resulted from shifting the window typically around 30 to 70 percent of the frame length [23]. If the weighting of the window equals to 1, means the window is rectangle, the response of the multiplication will be equal to the signal [39].

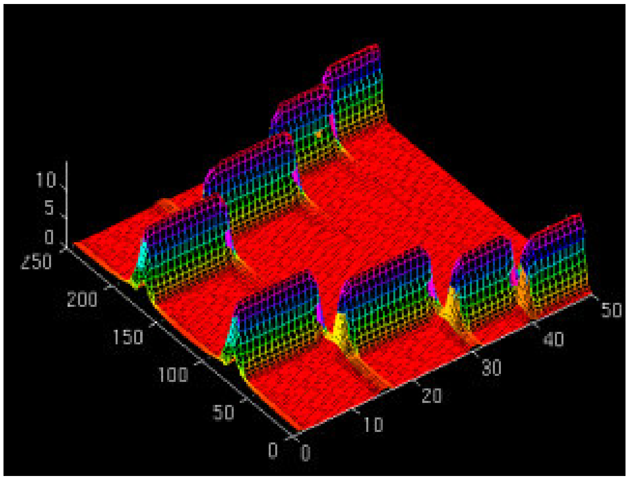


Figure 7: STFT transform of a non-stationary signal, with four frequency components at different times. The interval 0 to 250 ms is a simple sinusoid of 300 Hz, and the other 250 ms intervals are sinusoids of 200 Hz, 100 Hz, and 50 Hz, respectively [35].

2.2.2.2.1. Window Functions

The purpose of the windowing process is to enable the "*stationarity*" of the frequency components during the Fourier transform of the signal in time domain. Windowing in the time domain is multiplication of the frame and the window function and corresponds to convolution of the short-term spectrum with the magnitude response [23].

A good window function has narrow main lobe and low side lobe levels to reduce bias, in their transfer functions, unfortunately when the main lobe levels get narrower, side lobe levels increase as well [18,23,39].

2.2.3. Digital Filters

As a general description, a filter is a system that modifies the input signal into output signal [20,23,39,40]. In the time domain, filter is characterized by its *impulse response* that can be finite or infinite, despite that in frequency domain it can be specified by its *transfer function* H(z) where z is a complex variable[23,39].

2.2.3.1. Convolution Theorem

Linear Time Invariant (LTI) systems, which have the property

$$y[n-n_0]=T\{x[n-n_0]\} \qquad 2.4$$

can be characterized by their response to a unit impulse. Impulse response is usually denoted by h(n) and LTI systems can be described in terms of the *convolution* between the input signal and impulse response [20,23,36,39,40].

$$y[n]=\sum_{k=-\infty}^{\infty} x[k]h[n-k] \qquad 2.5$$

From the equation above, for each y[n] we sum the product of all values of x and time reversed impulse response whose position is determined by n [40]. Recursive relationship of this implementation is described by Ifeachor et. al. as follows where the coefficients a[k] and b[k] are determined from the filter specifications. The latter sum of the equation below represents the feedback part of the filter and equals zero for all k for finite impulse response filters [23].

$$y[n] = \sum_{k=0}^{N} a[k]x[n-k] - \sum_{k=1}^{M} b[k]y[n-k] \qquad 2.6$$

When a signal is periodic, equation 2.5 is undefined as the sum would be infinite, *circular convolution*, which sums over one period only, is used instead.

$$x[n] \otimes h[n] = \sum_{k=<N>} x[k]h[n-k] \qquad 2.7$$

where <N> represents any period of x [40]. As it can be seen from the equation 2.7, *finite impulse response filters*, FIR, use only current and the previous inputs non-recursively, despite that the *infinite impulse response filters*, IIR, use additionally to the current and the previous inputs previous outputs recursively [39].

2.2.4. Filterbanks, Sub-band Processing

Filterbanks, the term *sub-band processing* can be used instead, refer class of methods utilize the advantageous portions of separating the signal into different frequency ranges called sub-bands.

Another approach similar to sub-band processing is *Wavelet Transform* which is widely used in speech or non-speech processing instead of STFT [1,23,35]. This topic will be worked out with much care and great detail in further sections.

2.2.4.1. Filterbanks In Time Domain Analysis

A filterbank can be designed by using a set of recursive equations (2.6) in time domain. Analysis of the signal can be done for each sub-band signal using the regular frame processing. No matter the count of the sub-band is, analysis can be processed by using the same operations as in full band analysis [8].

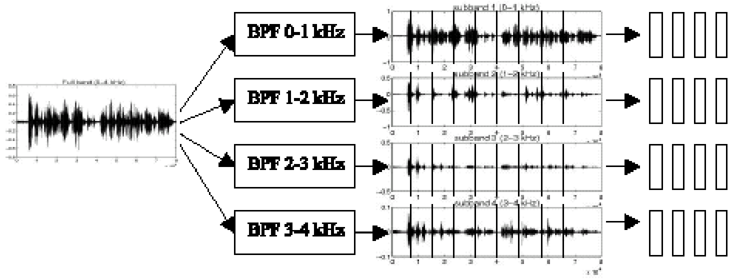


Figure 8: Sub-band signal analysis illustration in time domain analysis [24].

2.2.4.2. Filterbanks In Frequency Domain Analysis

In frequency domain, the filterbank can be expressed as multiplication of the signal spectrum with magnitude response. For a signal processed by M channel filterbank, output of the i'th filter of the filterbank can be expressed as

$$Y[i] = \sum_{j=1}^{N} X[j] H_i[j] \qquad 1 \le i \le M \quad 1 \le j \le N \tag{2.8}$$

$X[j]$ is j' th element of N point magnitude spectrum of the signal and $H_i[j]$ is the magnitude response of j' th element of i' th channel of M channel filterbank.

As a desired property, sum of the magnitude response of the filterbank at every frequency band must be equal to unity, which means that

$$\sum_{i=1}^{M} H_i[j] = 1 \qquad 2.9$$

for all j. This property enables to process each frequency band with the equal meaning. In the filterbanks that have uniform spacing in frequency scale, it might be difficult to ensure this property. At this time, *frequency warping* comes into the stage in order to adjust the resolution around a certain frequency [39]. Mel-, Bark- and ERB scales are some examples of *non-linear* functions that use frequency warping. Mel scale is the *non-parametric* frequency warping function and will be discussed later in feature extraction chapter.

2.2.5. Pre-Emphasis

Pre-emphasis refers to the filter that emphasizes higher frequencies of the spectrum. Numerical stability of the Linear Predictive analysis is inversely proportional to the dynamic range of the spectrum [29]. Therefore, a filter that flattens the spectrum should be used before Linear Predictive analysis.

Also some burst frequencies can be emitted by the target platform during the LOFAR analysis phase and those pup-ups may contain useful information about target identity. Since the lower frequencies of the spectrum contain more energy than the high ones, it might be useful to emphasize the higher frequencies of the spectrum.

The $\alpha > 0$ parameter controls the slope of the filter and typically selected between 0.96 – 0.99. By the convolution theorem the impulse response of the filter is implemented as a first order differentiator [7,29].

$$y[n] = s[n] - \alpha s[n-1] \qquad 2.10$$

2.2.6. Wavelet Transform

Wavelet analysis is one of the most promising data analysis technologies and used in a wide range of computational signal analysis activities. In this section we will cover the most relevant issues of using wavelets for some data processing techniques, like signal de-noising, data compression superficially, pattern detection and the spectral analysis of the signal components. Other applied fields that are making use of wavelets include astronomy, acoustics, nuclear engineering, signal and image processing, neurophysiology, music, magnetic resonance, imaging, optics, fractals, turbulence, earthquake-prediction, radar, human vision, and pure mathematics applications such as solving partial differential equations [17].

Wavelet transform has been used for many applications due to the capabilities of dealing with *non-stationary* and *linear time variant* signals and became the major data processing tool in many of the real life applications. Fourier transform was the major processing tool before the exploration that Wavelet Transform is more adequate for non-stationary signals. As we mentioned before, although FT is optimal as to the many criteria in case of stationary process, FT loses localization in time domain in case of non-stationary signal process. In order to recover this problem, STFT was used instead but the disadvantage of such approach is the high computational complexity of the decomposition algorithm.

In wavelet transform, instead of harmonic orthogonal functions, the framework consists of the functions generated by shifts and the compression functions in frequency and time domain are used. That allows dividing signal into details in frequency domain impeding loss of time localization [17,35].

If we summarize the wavelet transform, we process the time domain signal with various high-pass and low-pass filters that discriminate either high frequency or low frequency portions of the signal. This procedure is repeated, every time the signal is separated into the parts corresponding to some frequencies [35]. This operation is called *decomposition.*

Assuming that we have a signal, which has the frequency band 0-N Hz. First we split this band by passing the signal through low-pass and high-pass filters. So we have two portions of the signal with frequency bands 0-N/2 and N/2-N Hz. Parts. Now split the low-frequency portion (0-N/2 Hz.) into two parts again with the same process. This decomposition continues until reaching a predefined level.

At the end of the decomposition we have a bunch of signals, which actually represent the same signal, but all corresponding to different frequency bands. Examining the bunch of signals, we have the information about which frequencies exist at which time but according to the uncertainty principle originally proposed and formulated by *Heisenberg*, we cannot exactly know what frequency exists at what time instance, but we can only know what frequency bands exist at what time intervals.
The *uncertainty principl*e states that, the momentum and the position of a moving particle cannot be known simultaneously [7,35].

Main property of the Wavelet Transform, which is arisen from uncertainty principle, is resolution problem. We cannot determine the time information and the frequency information at a certain point together. Higher frequencies are better resolved in time, and lower frequencies are better resolved in frequency [35]. This means that, a certain high frequency component can be located well in time than a low frequency component. Accordingly, a low frequency component can be located well in frequency.

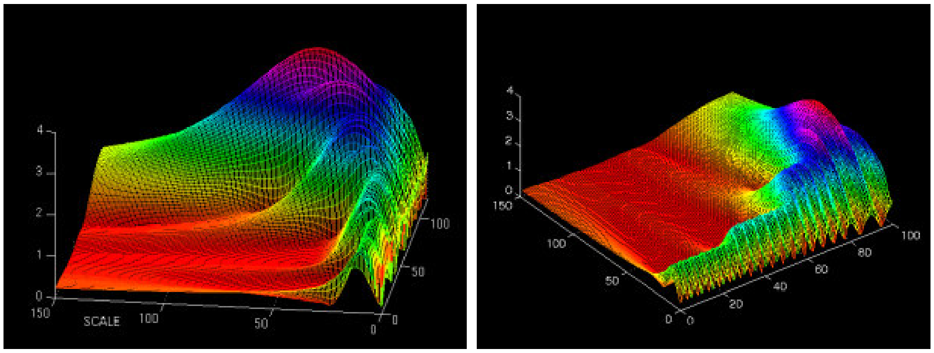


Figure 9: A continuous wavelet transform plot from Polikar [35]

2.2.6.1. Multi-Resolution Analysis

MRA can be likened to an observation about an unexplored object you have just found. First you take a fast look after that inspect with a microscope for the details. Make a research roughly and inspect with a magnifying glass again.

As was discussed earlier, unlike the STFT that provides uniform time resolution for all frequencies the Wavelet transform provides high time resolution and low frequency resolution for high frequencies and high frequency resolution and low time resolution for low frequencies [47]. This approach is very handy especially when the signal has high frequency components for short durations and low frequency components for long durations. Fortunately, the LOFAR signals emitted from the target platforms can be considered in this type as you see below.

Due to the signal's low-frequency components can be localized in the frequency domain and high-frequency components in the time domain the relation between the time and the frequency plane can be shown as below.

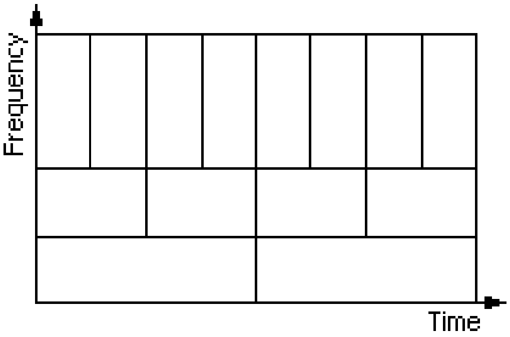


Figure 10: Time-Frequency plane in MRA

If we take a close look to the figure above, all the boxes in the plane have the same area although they have different width and height values. This relation implies that in low frequencies, height of the boxes get shorter despite that width of them gets larger, we have good frequency resolution and poor time resolution. In the upper part of the plane the relationship has reverse direction [17,35].

2.2.6.2. Wavelets

The most important component of the wavelet transform is, absolutely, the wavelet, which acts like a window and serves as prototype.

Wavelet transforms have an infinite set of basis functions, although Fourier Transforms have just the sine and cosine functions. Thus wavelet analysis provides immediate access to information, which cannot be provided by other time-frequency methods. It differs how compactly the basis functions are localized in space and how smooth they are, in different wavelet families [17].

Some of the wavelet families have a fragmented geometric shape that can be subdivided in parts, each of which is a smaller copy of the whole and these wavelets are generally self-similar and independent of scale (they look similar, no matter how close you zoom in). Daubechies filter is one of the examples of that kind.

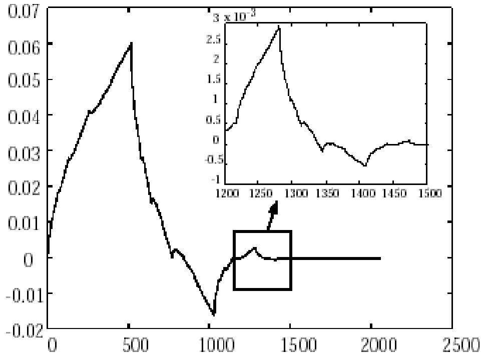


Figure 11: The self-similarity of the Daubechies mother wavelet. Generated using the WaveLab. The inset figure was created by zooming into the region x =1200 to 1500 [17].

Another important properties of the wavelets are admissibility and regularity conditions, which give their names to the functions we mentioned about [49].

It can be said [43,49] that square integrable functions $\psi\ (t)$ satisfying the *admissibility condition*,

$$\int \frac{|\Psi(\omega)|^2}{\omega} \mathrm{d}\omega < +\infty \tag{2.11}$$

can be used to first analyze and then reconstruct a signal without loss of information. $\Psi(\omega)$ stands for the Fourier transform of $\psi(t)$. The admissibility condition implies that the Fourier transform of $\psi(t)$ vanishes at the zero frequency,

$$|\Psi(\omega)|^2 \Big|_{\omega=0} = 0 \tag{2.12}$$

A zero at the zero frequency also means that the average value of the wavelet in the time domain must be zero,

$$\int \psi(t)dt = 0 \qquad 2.13$$

and therefore it must be oscillatory. In other words, $\psi(t)$ must be a wave [49]. Detailed information about the mathematical background can be found in Valens and Polikar.

Within each family of wavelets, the wavelet subclasses are distinguished by the number of coefficients and by the level of iteration. Wavelets are classified within a family most often by the number of vanishing moments. For example, within the Coiflet wavelet family there are Coiflets with two vanishing moments, and same Coiflets with three vanishing moments [17].

2.2.6.3. Discrete Wavelet Transform

Although CWT provides us useful information, which cannot be provided by different transformations about the signal, it has some properties that make it difficult to use directly. The first is the redundancy. As we knew before, the wavelet transform is calculated by continuously shifting a scalable function over a signal all the times and by calculating the similarity between the wavelet and the signal. It will be clear that the obtained wavelet coefficients will therefore be highly redundant [49].

Even without the redundancy of the CWT we still have an infinite number of wavelets in the wavelet transform and we would like to see this number reduced to a more manageable count [49].

As another problem is that for most functions the wavelet transforms have no analytical solutions and they can be calculated only numerically or by an optical analog computer [49].

CWT doubles the signal dimension due to time-scale representation so that is highly redundant. To overcome this problem *discrete wavelets* have been introduced [9,49]. Discrete wavelets are not continuously scalable and translatable but can only be scaled and translated in discrete steps.

$$\psi_{j,k}\left[t\right]=\frac{1}{\sqrt{s_0^j}}\psi\left[\frac{t-k\tau_0 s_0^j}{s_0^j}\right] \qquad 2.14$$

DWT coefficients are usually sampled from the CWT on a *dyadic grid*, choosing s_0 = 2 and τ_0 = 1. Although we removed the redundancy, we should reduce the number of the wavelets. If one wavelet can be seen as a band-pass filter, then a series of dilated wavelets together with a scaling function can be seen as a band-pass filter bank [49]. If we compare the center frequency of the wavelets with width of them, we see the ratio between them is the same. By setting this ratio, this peculiar filter can be designed.

If we consider Wavelet Transform as a filterbank processing, the signal is passed through a series of high pass filters to analyze the high frequencies, and it is passed through a series of low pass filters to analyze the low frequencies. Mallat was the first to implement this scheme, using a filter design called *sub-band coding*, yielding a *Fast Wavelet Transform*. Also a technique very similar to sub-band coding that is known as Multi-resolution Analysis (a.k.a. *pyramidal coding*) is described in [35].

Splitting the signal spectrum in two parts, a low-pass and a high-pass part, is a way to design the filterbank. The high-pass output part contains the smallest details, low scale, high frequency components we are interested in. However, the low-pass part still contains some details and therefore we can split it again. Also low-pass part output contains approximations, which means high scale - low frequency components. We split the approximations we have again, until we are satisfied with the number of bands we have created. In this way we have created an *iterated filter bank*. Usually the number of bands is limited by for instance the amount of data or computation power available.

While changing the resolution of the signal, which is a measure of the amount of detail information in the signal, by filtering operations, the scale is changed by up-sampling and down-sampling operations (*decimation*). Down-sampling a signal corresponds to reducing the sampling rate, the Multi-resolution nature (*j* parameter in Eq. 2.14) of the scaling functions.

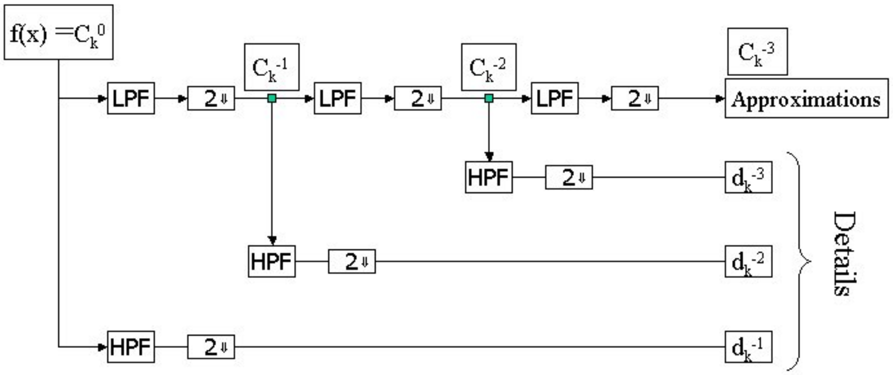


Figure 12: Pyramidal structure of DWT

As was indicated in equation 2.7, filtering a signal corresponds to the mathematical operation of circular convolution of the signal with the impulse response of the filter in discrete time. In the pyramidal structure above we have a down-sampling operation, which can be described mathematically as

$$y_{high}(k) = \sum_{n} x(n) g(2k - n)$$
$$y_{low}(k) = \sum_{n} x(n) h(2k - n) \qquad 2.15$$

Where $y_{high}(k)$ and $y_{low}(k)$ represents the outputs of the filters [35,49].

If we mention about the synthesis, the analysis procedure can be followed in reverse direction for the sake of reconstruction of the original signal from decompositions. This procedure is also easy because of the relationship between the low-pass and high-pass filters.

However, if the filters are not ideal half-band, then perfect reconstruction cannot be achieved. Although it is not possible to realize ideal filters, under certain conditions it is possible to find filters that provide perfect reconstruction, i.e. *Daubechies*' wavelets developed by I. Daubechies. That means if we want to reconstruct the original signal form decompositions accurately, we cannot choose an arbitrary wavelet.

Due to successive down-sampling operation, the signal length must be power of two, or at least a multiple of power of two. The length of the signal determines the number of levels that the signal can be decomposed [9].

CHAPTER 3

FEATURE EXTRACTION

Feature extraction is the most discriminative part of the process of recognition and classification scheme. Feature extraction is a process, which eliminates the relevant features from raw data containing too much irrelevant information about the target and has the goal of representing the complete class by the information vector emphasizing the class distinguishability and signature. Finally, extracted features should contain some unique characteristics that are effective in class discrimination, be more stable, robust, compact representation than raw signal and be small size suitable to use with further classification or recognition applications.

Many statistical and signal processing methods can be used for this purpose. In this study, basic feature extraction methods used in speech and music classification were investigated.

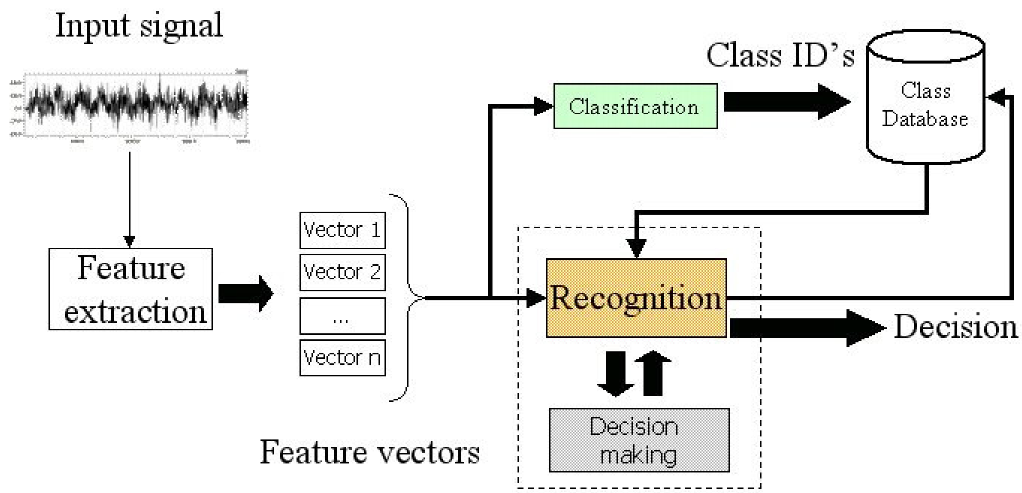


Figure 13: General components of acoustic signal processing system.

In the general diagram of the system, classification represents the training phase of the core database management system and also recognition represents a decision making process that is using trained database and decision logic together. Both parts of that system need feature extraction phase before so they use sequence of numerical descriptors into which the acoustic signal is converted by this sometimes called *front-end* of system.

Formally, converting originally high dimensional vectors to the low dimensional vectors is considered as feature extraction. It is important to reduce the vector dimension into lower values because of the phenomenon *curse of dimensionality*, which states that the amount of training samples grows exponentially with the dimensionality [23,40]. And also decreasing the dimension of the raw data vector decreases the computational complexity as well. Typically front end can be represented as a mapping $f:\mathbb{R}^N \to \mathbb{R}^d \qquad d << N$ formally [24].

According to [23,41] optimal feature sets should have the following properties.

1. High between class variation,
2. Low within class variation,
3. Easy to measure,
4. Robust against environmental effects and propagation losses.
5. Robust against distortion and noise,
6. Maximally independent of the other features.

As it can be understood above the first two properties, ensuring high between and low within class separation provides main clues about signal discrimination. It is very hard issue to ensure both and sometimes they have some side effects. If one assumes that two signals recorded from the same target but operating different auxiliaries in each record, high between-class separation causes those signals have different emitters in spite of that the emitter is just the same.

The feature should be constructed from the short segments of the signal [24] and on the trot. That means that the feature should occur frequently in complete signal. For the reason that it is not possible to exempt the aid of a human expert from feature extraction of underwater acoustic signal classification system, this process should be as easy as doing this automatically and this states that features should be *easily measurable*.

Features extracted from a specific signal should also be *not susceptible* to change of environmental conditions in where the signal is recorded. Also each vector of the feature set, namely *codebook*, should be *maximally independent of each other*. If two correlated features are combined, nothing is gained, and in fact, this may even degrade recognition results.

In the figure below, there are two pictures generated by author's software. In the figures every class represents set of features, which belongs to specific signal and each dot represents a feature vector. The vectors plotted in three dimensions although all of them have the dimension of 32. The figures are just same except the view angles.

Codebooks were extracted from a UN type vessel data recorded while in conditions “B” and “Sn”. The blue and the gray dots represent “Sn” and the others represent the “B” condition. It is obvious that the features ensure high between-class separation, “Sn – B”. The distance between the classes is high enough. In spite of the distinction between the propulsion related classifications, the within-class separation is as low as possible in propulsion related classifications.

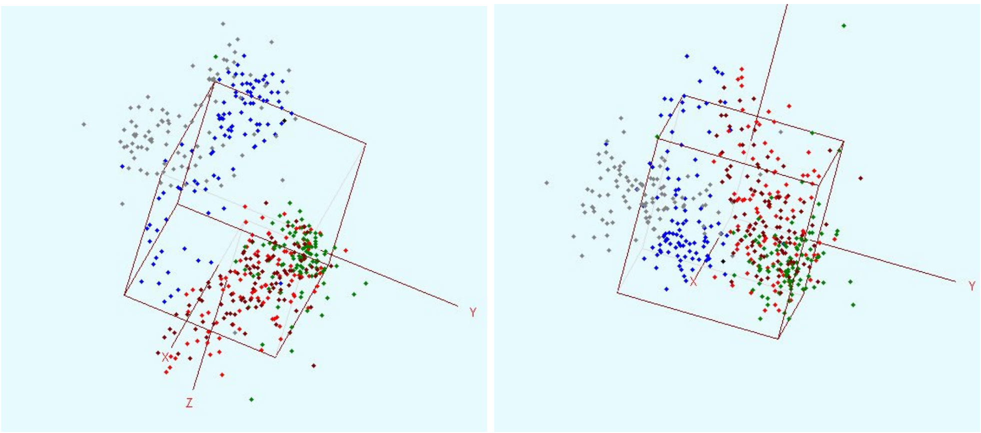


Figure 14: 3D Feature vector plots of UN class vessel.

There are three types of feature extraction schemes.

1. Frequency domain based (FFT) feature extraction.
2. Filterbank based feature extraction.
3. Wavelet based feature extraction.

3.1. Cepstral Analysis

The cepstrum was defined by Bogert et. al.[2] in 1963 and it is a process of taking FFT of a decibel spectrum of a signal as if it were a signal. Cepstrum verbally is an anagram of spectrum.

Spectral analysis is very good at representing sound signals but there are some drawbacks also. *Deller* proposes that spectral analysis is useful in representation, provides valuable intensity information in time-frequency cells but weak at phase and fine – scale timing information productivity. At this point the cepstral analysis becomes starting point of other heuristic methods of signal representations such as LPCC, MFCC.

Basically cepstral analysis in sound recognition, especially in speech, is used for parametric representation of the envelope structure (caused by resonance of vocal tract in speech) of the short-term spectrum of the signal [10,24]. The analysis process can be described as d*econvolution* of the signal, which separates the excitation signal from filter [32], e.g. resonance filter in speech.

As we know from DSP filter background, perceived signal is typical convolution of excitation signal $e(n)$ and filter $h(n)$ where $s(n) = e(n) \otimes h(n)$. Purpose of the cepstral analysis is to separate $e(n)$ and $h(n)$.

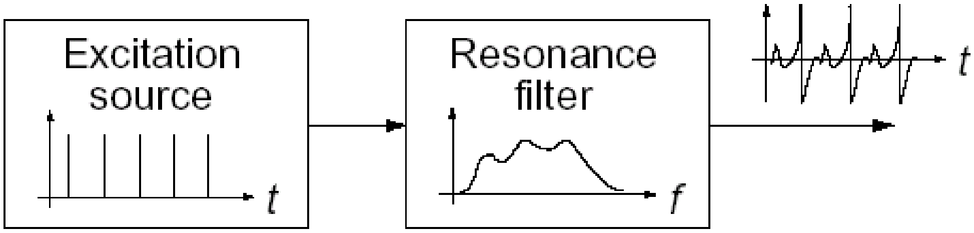


Figure 15: Convolution of the signals generating perceived signal [11].

Cepstral analysis procedure (deconvolution) can be explained as follows:

Convolution ➔ $s(n) = e(n) \otimes h(n)$

FFT Product ➔ $G(e^{j\omega}).H(e^{j\omega})$

Log sum ➔ $\log G(e^{j\omega}) + \log H(e^{j\omega})$

Homomorphic deconvolution ➔ $c_g(n) + c_h(n)$

Finally Deller defines the real cepstrum as follows [10,23,24,32].

➔ $c_n = IDFT(\log\left|DFT(s(n))\right|)$

Below the figure, the block diagram of a typical cepstral coefficients extractor is shown.

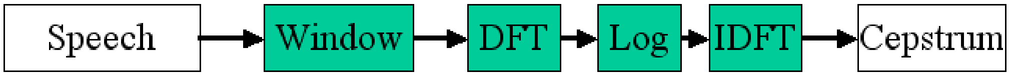


Figure 16: Cepstral feature extractor for speech.

By keeping only the first 10 to 20 components of the real cepstrum, called the cepstral coefficients, it will be enough to estimate an approximation to the transfer function with fewer coefficients because higher frequency components will be discarded [33].

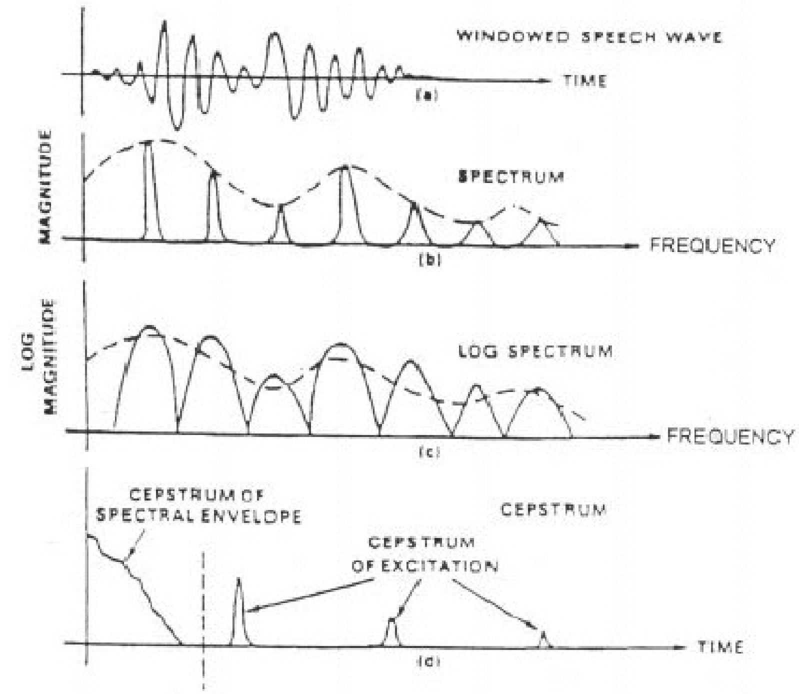


Figure 17: Illustration of Cepstral Analysis for speech signal [32].

3.2. FFT Based Feature Extraction

As we mentioned before, cepstral analysis is the starting point of many feature extraction methods. The two common choices for extraction of cepstral coefficients are based on a filterbank model and a linear predictive (LP) model.

3.2.1. Linear Prediction

3.2.1.1. Time Domain Interpretation

In time domain, adjacent samples of an acoustic signal are highly correlated so the general signal behavior can be estimated by inspecting a few past samples of it. Also it will not be possible to predict complete signal behavior so we have to consider a prediction error. The aim of linear prediction is, basically computation of prediction coefficients by minimizing the squared prediction error. LP model assumes that each sample can be approximated by a linear combination of a few pas samples [23].

$$
\begin{aligned}
s(n) &= s'(n) + e(n) \\
s'(n) &= \sum_{k=1}^{p} a_k s(n-k)
\end{aligned}
\qquad 3.1
$$

In the equation 3.1, p represents the order of the predictor. The goal of the model is to compute the predictor coefficients $\{a(k) | k = 1, 2, ..., p\}$ ensuring the minimal square prediction error E as small as possible.

$$
\begin{aligned}
E &= \sum_n \|e(n)\|^2 \\
&= \sum_n \left(s(n) - \sum_{k=1}^{p} a(k) s(n-k) \right)^2
\end{aligned}
\qquad 3.2
$$

To find the minimum value, the partial derivatives of E with respect to the predictor coefficients $\{a(k)\langle k = 1, 2, ..., p\rangle\}$ are set to zero.

$$
\frac{\partial E}{\partial a_k} = 0 \qquad 3.3
$$

If we write out the expression for all p values of k, problem of finding the optimal predictor coefficients reduces in solving Yule-Walker equation namely AR equation.

There are two methods to solve AR equations, autocorrelation method, and covariance method. In this section we will inspect the former one. Differentiation and discretization of the equation 3.3 can be expressed as follows.

$$\sum_n 2\left[s(n)-\sum_{j=1}^{p} a_j s(n-j)\right].[-s(n-k)]=0$$

$$\sum_n s(n)s(n-k)=\sum_j a_j.\sum_n s(n-j)s(n-k)$$

$$\Phi(0,k)=\sum_j a_j.\Phi(j,k)$$

$$\left\{\Phi(j,k)=\sum_{n=1}^{m} s(n-j)s(n-k)\right\} \tag{3.4}$$

$\Phi(j,k)$ represents the autocorrelation coefficients and finally we have p linear equations to solve all a’s. If we write out the matrix form,

$$\underset{r}{\begin{bmatrix} r(1) \\ r(2) \\ \cdots \\ r(p) \end{bmatrix}} = \underset{R}{\begin{bmatrix} r(0) & r(1) & \dots & r(p-1) \\ r(1) & r(2) & \dots & r(p-2) \\ \cdots & \cdots & \cdots & \cdots \\ r(p-1) & r(p-2) & \dots & r(0) \end{bmatrix}} \underset{a}{\begin{bmatrix} a_1 \\ a_2 \\ \cdots \\ a_p \end{bmatrix}}$$

we have $Ra = r$ type equation. In that case $Ra = r$ is a special type of matrix, which is called Toeplitz matrix. The vector a represents the LP coefficients and r vector represents the autocorrelation sequence.

$$\Phi(j,k)=\sum_n s(n-j)s(n-k)=r(|j-k|) \tag{3.5}$$

There exists an efficient algorithm to find the solution for AR equations [24], *Levinson-Durbin recursion*. This algorithm takes signal and autocorrelation sequence as input and gives predictor and reflection coefficients as output.

If the original spectrum has a wide dynamic range then LP model may become unsteady for those. At that time it will be useful to apply a pre-emphasize algorithm [29], which whitens the signal and reduces the dynamic range to the signal before application of LP model [23,24].

A schematic diagram of a LP Model application is shown below.

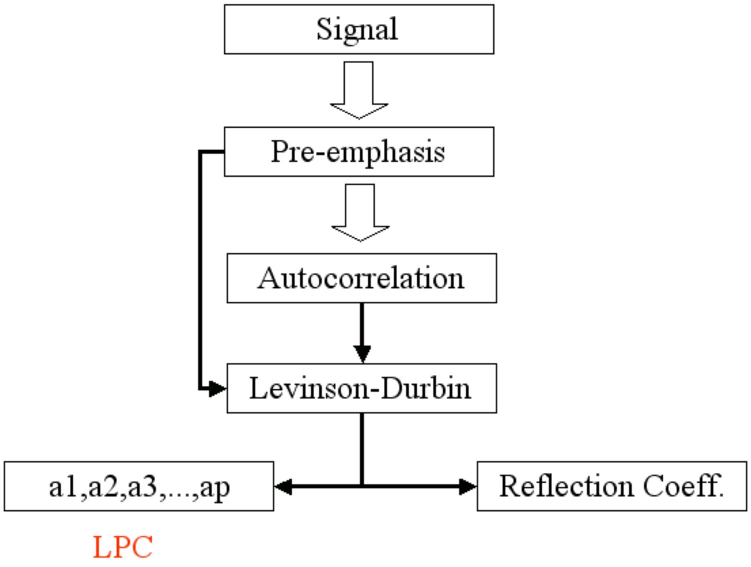


Figure 18: Typical LP Model

3.2.1.2. Frequency Domain Interpretation

A signal can be approximated with the LP model with a small prediction error depending on the LP order. Also optimal LP order is relevant to what kind of information we want to extract from spectrum. Makhoul [29] has proved that minimization of square prediction error is equal to minimizing the square error between signal magnitude spectrum and the LP magnitude response, that means resulting transfer function from LP is a least square approximation of the original magnitude spectrum [23].

In frequency domain, the emitted signal can be expressed by the convolution of excitation signal with an IIR filter kernel and the transfer function can be given by:

$$H(z) = \frac{S(z)}{E(z)} = \frac{1}{(1 - \sum_{k=1}^{p} s(k) z^{-k}} = \frac{1}{A(z)} \qquad 3.6$$

The filter has no zeros so this filter is called all-pole filter [23,29,10,37].

As it can be seen in *Figure 19*, decreasing the prediction order smoothes the original spectrum and increment of p causes the LP model to fit the original spectrum.

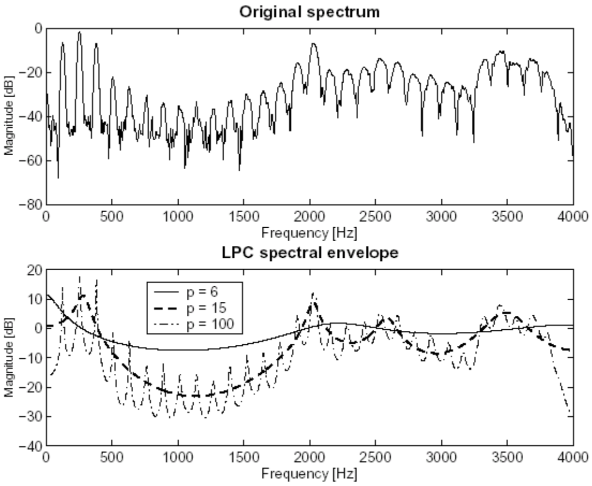


Figure 19: Estimation of the spectral envelope by LP analysis using different order predictors [23].

A practical rule about prediction order was represented in [19] for efficient model estimation. One pole per each kilohertz plus 2 – 4 poles. Since the complex poles must be real or occur in complex conjugate pairs to ensure that the filter coefficients are real, the model order is twice the number of poles. At that point we should notice that this rule is designed for *speech* recognition purposes.

3.2.2. Linear Predictive Cepstral Coefficients

Although it is possible to use LP Coefficients directly in representing the signal features, in practice it is observed [23] that the adjacent predictor coefficients are highly correlated. It is obvious that less correlated representations would be more efficient.

Cepstral coefficients derived from magnitude spectrum are a little bit different from the cepstral coefficients derived from LP cepstral coefficients since those are derived directly from predictor coefficients. Given the LP coefficients $\left\{ a(k) \middle| k = 1, 2, \ldots\ldots\ldots, p \right\}$, cepstral coefficients are computed by the following recursive formula [23].

$$c(n) = \begin{cases} a(n) - \sum_{k=1}^{n-1} \frac{k}{n} c(k) a(n-k) & 1 \leq n \leq p \\ -\sum_{k=1}^{n-1} \frac{k}{n} c(k) a(n-k) & n > p \end{cases} \qquad 3.7$$

3.3. Filterbank Based Feature Extraction

Mainly, filterbanks are used to emphasize some of the frequency portions in power spectrum of the sound signal like ear does which is also called perceptual weighting.

More filter banks process the spectrum below 1 kHz [25] since the speech signal contains most of its useful information in lower frequencies. In order to emphasize the specific portions of the spectrum, different frequency warping methods are used instead of linear frequency warping during cepstral analysis. Mel [7,25,34,37,40] is one of the most widely used frequency warping methods in signal processing, especially with speech signals.

3.3.1. Mel Frequency Cepstral Coefficients (MFCC)

Instead of linear warping methods, some logarithmic warping methods are used during spectral analysis. Basically most of the information characterizing the signal is contained in lower frequency contents of the spectrum. Human perception system, which space the lower frequency channels linearly and higher frequency channels logarithmically mainly motivates MFCC method in feature extraction.

Mel equivalent of linear frequency axis can be computed by the following formula.

$$mel(f) = 2595.\log(1+\frac{f}{700}) \qquad 3.8$$

The reverse transformation can be achieved by

$$f(mel) = 700(e^{\frac{mel}{1125}} - 1) \qquad 3.9$$

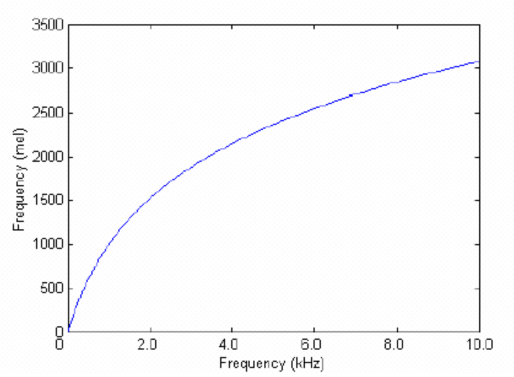


Figure 20: Mel scale

In the first step of MFCC, original signal spectrum is divided into segments and frequency spectrum is computed via FFT. After this process, linear spectrum values are passed through a special filterbank, which consists of triangular-shaped filters that emphasize center frequency ω_i and span to the next center frequency [40].

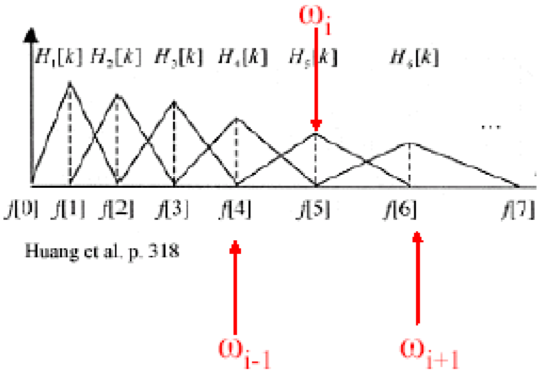

ω_i
$H_1[k]$ $H_2[k]$ $H_3[k]$ $H_4[k]$ $H_5[k]$ $H_6[k]$
...
$f[0]$ $f[1]$ $f[2]$ $f[3]$ $f[4]$ $f[5]$ $f[6]$ $f[7]$
Huang et al. p. 318
ω_{i-1}
ω_{i+1}

Figure 21: Mel Filterbank [40].

While constructing the filterbank, we should consider that the area under each filter must be constant. As it can be seen from *Figure 21*, x-axis of the filterbank is linear frequency scale but the shape of the triangular filters are achieved by distributing the filters uniformly across the mel scale [40].

$$H_m[k]=\begin{cases} 0 & k<f[m-1] \\ \dfrac{2(k-f[m-1])}{(f[m+1]-f[m-1])(f[m]-f[m-1])} & f[m-1]\le k\le f[m] \\ \dfrac{2(f[m+1]-k)}{(f[m+1]-f[m-1])(f[m]-f[m-1])} & f[m]\le k\le f[m+1] \\ 0 & k>f[m+1] \end{cases}$$

After construction of the filterbank, the above can compute frequency response of the filter, just algebraic, formula, where m represents the channel number, k represents the frequency line, $f(m)$ represents the frequency bin associated with m th center frequency and $H_m[k]$ represents the filter response. Complete filterbank response can be computed for each channel $m=1,......,M$ by the following formula [40].

$$S(m)=\log\left\lfloor \sum_{k=0}^{N-1} X^2(k)H_m(k) \right\rfloor \quad 0<m\le M \qquad 3.10$$

Finally taking Discrete Cosine Transform of S vector [40], instead of IDFT, can derive MFCC coefficients. Since the log power spectrum is symmetric and real, the inverse DFT reduces to discrete cosine transformation (DCT). This transformation makes the features uncorrelated, which leads using diagonal covariance matrices instead of full covariance matrices while modeling the feature coefficients by linear combinations of Gaussian functions. Therefore complexity and computational cost can be reduced. Since DCT gathers most of the information in the signal to its lower order coefficients, by discarding the higher order coefficients significant reduction in computational cost can be achieved [23,25].

$$c(n)=\sum_{m=0}^{M-1} S(m)\cos\left(\frac{\pi n(m-\frac{1}{2})}{M}\right) \quad 0\le n<M \qquad 3.11$$

$c[0]$ corresponds to the energy level of the spectrum (log magnitude) and it depends on the intensity, so it is excluded from feature vector for this reason. In speech processing, especially 12 coefficients except $c[0]$ of 24 channels are used. The most important property of cepstral coefficients derived by MFCC is somewhat; they are uncorrelated with each other.

Also dynamic features of the signal can be imported into the feature sets as cepstral coefficients. The meaning of dynamic features is the 1st and the 2nd derivatives of cepstral coefficients. Those derivatives can be taken into account as delta and delta-delta cepstral coefficients. In this section, this kind of derivation will not be covered.

3.4. Wavelet Based Feature Extraction

A generalization of Wavelet transform originally designed mainly for signal compression is the *Best Basis* algorithm of Coifman and Wickerhauser [5]. This method first expands a given signal or a given collection of signals into a library of orthonormal bases i.e. a redundant set of wavelet packet bases or local trigonometric bases having a binary tree structure where the nodes of the tree represent subspaces with different time – frequency localization characteristics. Then a complete basis called a best basis, which minimizes a certain information cost functional τ is searched in this binary tree using the divide and conquer algorithm [42].

The cost functional τ should describe the concentration or the number of coefficients to accurately describe the sequence for practical reasons [5].

In summary, the aim is to extract the maximum information or *features* from our signal by projection onto a co-ordinate system or basis function in which that signal is best represented. In classification problems, efficiency means that, a basis, which most uniquely represents a given class of signal in the presence of other known classes, will be most desirable [27].

More illustrative description of Best Basis Algorithm has been elucidated in [42] according to [5] as follows. It is assumed that we have a vector x,

Step 0: Choose a time frequency decomposition method i.e. specify wavelet packet transform as a pair of quadrature mirror filters (i.e. Daubechies), local cosine transform, or local sine transform.

Step 1: Expand x into the library of orthonormal bases (binary tree like) and compute the coefficients:

A set of Basis vectors belonging to the relevant subspace,
Best Basis for the signal x.

Step 2: Determine the best subspace using the cost functional τ, which is an additive for high-speed computations.

Saito proposed Local Discriminant Basis algorithm for the sake of obtaining the suitable basis in the library of orthonormal bases, shortly WPD tree. It is similar to the Best Basis algorithm, which is designed for data compression, however in LDB a basis selection criterion, which most uniquely represents a given class of signal in the presence of other known classes, is designed. Saito and Coifman [42] explained the LDB algorithm in following steps. Assumed that we have a training data set L consisting of K classes. $\left\{\left\{x_i^{(k)}\right\}_{i=1}^{N}\right\}_{k=1}^{K}$

Step 0: Choose a time frequency decomposition method i.e. specify wavelet packet transform as a pair of quadrature mirror filters (i.e. Daubechies), local cosine transform, or local sine transform.

Step 1: Construct time frequency energy maps Γ_k for *k = 1…K*. Original LDB node selection algorithm based on the averaged pair – wise differences of the signal class energy distributions.

Step 2: Expand x into the library of orthonormal bases (binary tree) and compute the coefficients:

A set of Basis vectors belonging to the relevant subspace,
Best Basis for the signal x.

Step 3: Determine the best subspace using the additive discriminant measure $\Im$.

Step 4: Order the basis functions by their power of discrimination.

Step 5: Use most discriminative basis function to construct the classifier.

Detailed information, mathematical background, and theoretical descriptions about Best Basis and LDB can be found in [5,42].

Başaran et.al. [1] have reported that the validity of the selection criterion has become less meaningful as the number of the classes increases and also LDB scheme was quite sensitive to time synchronization issues. They considered that the reason of the low detection rate of the experiments based on LDB was the possible defectiveness about the time synchronization of the LDB scheme.

3.5. Distance Measures[4]

3.5.1. Euclidean Distance Function

Euclidean distance function measures the shortest distance between two distinct points x and y, as the crow flies. At that point it differs from Manhattan distance measure (block distance measure) since Manhattan measures the total length of the perpendicular edges. Distance between the points X and Y can be computed by the following equation.

$$d^2(x_i, y_j) = \sum_{k=1}^{D}(x_k^i - y_k^j)^2 = \left(x^i - y^j\right)^{\mathrm{T}}\left(x^i - y^j\right) \qquad 3.12$$

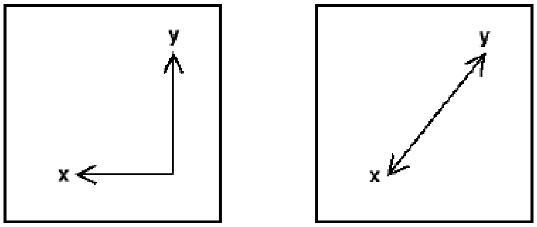


Figure 22: Illustration of the difference between two distance measurement functions. Manhattan and Euclidean Distance Function.

3.5.2. Squared Euclidean Distance Function

It is almost the same as the regular Euclidean except does not take square root. As a result, clustering with the Euclidean Squared distance metric is faster than clustering with the regular Euclidean distance. The output of Jarvis-Patrick and K-Means

[4] Some of the information and the all the figures in this section were taken from Predictive Patterns Software official web page, which is online at http://www.predictivepatterns.com/index.html.

clustering is not affected if Euclidean distance is replaced with Euclidean squared. However, the output of hierarchical clustering is likely to change.

In the experimental progress of the project, Euclidean Square Distance Function had the best performance over the other (computationally expensive) distance measurement methods when they have been used together with Kohonen SOM's or GMMs.

3.5.3. Weighted Euclidean Distance Function

In Weighted Euclidean Distance computation, each pair of the inputs are multiplied with a weighting variable so it can be defined similarly regular EDF as follows.

$$d_W^2(x_i, y_j) = \left(x^i - y^j\right)^{\mathrm{T}} W \left(x^i - y^j\right) \quad 3.13$$

where;

$$W = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$$

if $W = \Gamma^{-1}$ where Γ is the *covariance matrix* of the inputs then equation 3.13 represents the *Mahalanobis* distance which is nothing more than Weighted EDF.

Mahalanobis distance measurement is a very useful way of determining the similarity of a data set from an unknown sample to a data set measured from a collection of known samples, and also it takes distribution of the points (mean and standard deviation) into account [34].

3.5.4. Chebychev Distance Metric

Chebychev Distance Metric represents the maximum distance between two points in a single dimension and the distance can be computed as:

$$d_W(x_i, y_i) = \max_i |x_i - y_i| \qquad 3.14$$

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 together.

3.5.5. Pearson Correlation and Pearson Squared

Pearson Correlation measures the similarity between two profiles while the Pearson Squared measures inverse relationships additionally. Pearson Correlation function can be formulated as follows.

$$d(x_i, y_i) = 1 - \frac{\left(\frac{x_i - \mu_x}{\sigma_x} \bullet \frac{y_i - \mu_y}{\sigma_y} \right)}{n} \qquad 3.15$$

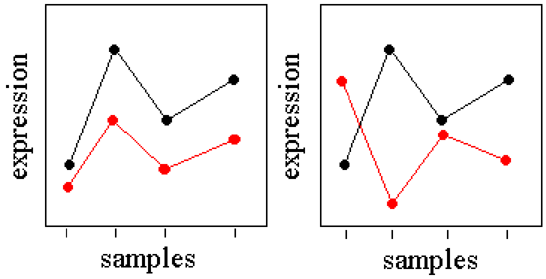


Figure 23: Similarity and inverse similarity of two profiles.

As it can be seen in the *Figure 23*, the data sets on the left hand side have closely perfect correlation but the latter sets are almost perfectly anti – correlated. So while Pearson Correlation and Pearson Squared place the left one into the same cluster, the right one will be placed into remote clusters.

CHAPTER 4

CLASSIFICATION AND RECOGNITION

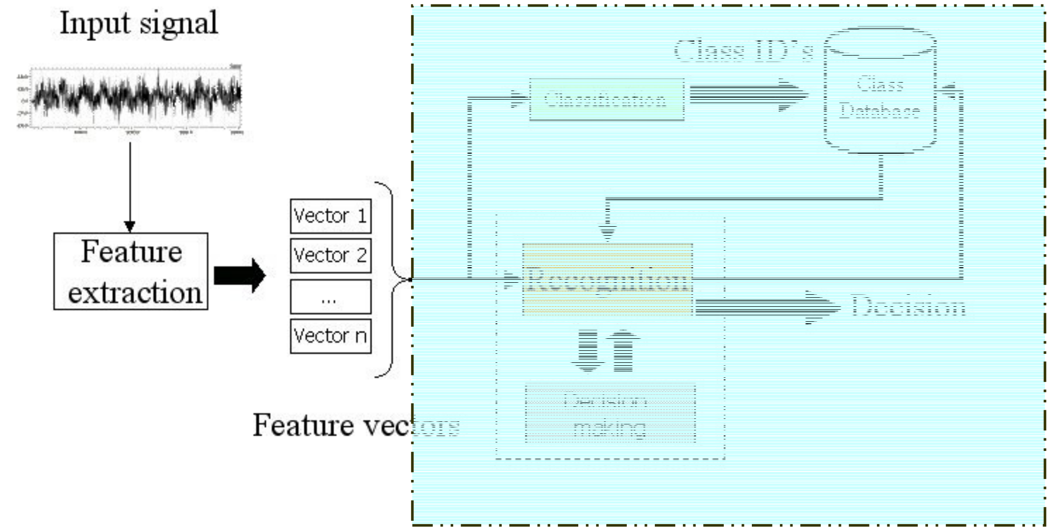


Figure 24: General components of acoustic signal processing system.

In this chapter, we will investigate the existing data classification and recognition algorithms that have been widely used in speech or non-speech signal processing operations. Detailed information about feature extraction techniques that have been used as data providers (feature vectors) for classificatory of the general acoustic signal processing system (especially speech) have been presented in last section.

The following classification techniques are commonly used for sound classification application domain. Those are:

- Dynamic Time Warping – DTW
- Hidden Markov Models – HMM
- Learning Vector Quantization – LVQ
- Kohonen Self Organizing Maps – SOM
- Artificial Neural Networks – ANN
- Gaussian Mixture Models – GMM

In this chapter, the last four classification techniques will be highlighted. Also the probable combinations of those techniques with the feature extraction schemes emphasized before will be discussed.

Cowling [7] states that any of the classification techniques using sub word features are not suitable for non-speech sound identification for the reason that there is no way to split the non-speech sounds into slices by using an alphabet. Due to the lack of an alphabet HMM based classification techniques will be hard to use in underwater acoustic sound classification schemes. A useful taxonomy development in underwater acoustic domain will lead new research areas but this is the beyond the purpose of this study.

The other classification techniques are related to each other. They are comprised of many common algorithms i.e clustering algorithms, distance measure techniques, and perceptual learning algorithms. Also some of them are complementary of each other.

Classification and the recognition terms carry different meanings in terms of the signal-processing domain. But also both operations have a process of clustering data sets somewhat collected by a feature extractor, in corresponding vector space according to the similarities or dissimilarities that the vectors have by using a purpose specific distance metric. On the other hand recognition has two different processes that can be used together or standalone, *Identification* and *Verification*. Both of those processes have a decision-making module giving the answer Yes / No for verification as well as presenting the identity of the target, which represented by the processed signal in identification.

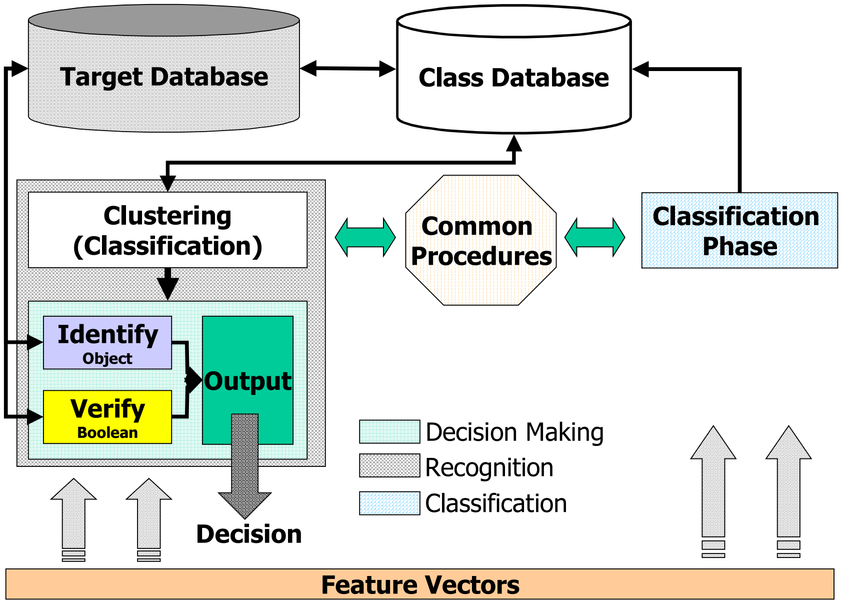


Figure 25: Basic classification and recognition processes.

Before investigating the classification techniques above, we will explore basic algorithms commonly used in Pattern Recognition area.

4.1. Clustering Problem

Clustering is combinatorial optimization problem of partitioning data vectors into clusters (represented by code vectors) each of them represents a particular group of data vectors and similar vectors are grouped together as dissimilar vectors are grouped in different clusters [15]. Clustering is the fundamental algorithm of vector quantization techniques, which have been widely used in codebook generation, image analysis, and data compression[5].

Formally, consider a set of N training vectors; we use the term *training vectors* for initial data vectors, in a K-dimensional space. Clustering is to find a codebook C of M code vectors by minimizing the distance (or given cost function) between the training vectors and their representative code vectors. Special instances of distance metrics can be used according to the vector properties and project conditionals.

Clustering issues strive for two main problems, how many clusters there are in the data set, and where they will be located. There are different approaches solving those problems according to the condition whether we know the exact cluster number or not. Fränti P. et al [15,21,22] splits the problem into two groups, *static clustering* if the number of clusters is known beforehand, and *dynamic clustering* if the number of clusters must also be solved. In this study, we will examine the static clustering techniques rather than the dynamic clustering. Generalized Lloyd Algorithm, Nearest Neighbor Algorithm [14], Genetic Algorithm, and Randomized Local Search [13] can be used in solving static clustering problem. Also there are some additional methods that can be found in [21].

Using heuristic algorithms like Agglomerative Clustering [16] or Genetic Algorithm as an evolutionary approach [30] can solve dynamic clustering problems. As an agglomerative approach, Stepwise clustering algorithm [21] tries to find solutions for

[5] A Java™ Applet demonstrating most of the clustering algorithms can be found at www.neuroinformatik.ruhr-uni-bochum.de\ini\VDM\research\gsn\DemoGNG online.

every step having cluster count (m, m+1, m + 2. … M_{max}) iteratively by using any other static clustering algorithms such as GLA, LBG-U, and RLS.

4.2. Continuous Density Models (Gaussian Mixtures)

There is another way to build a classifier based on a vector quantization method that does not rely on a codebook. This is called as continuous density model. To implement this, we need to define a probability density function over the training data set, in practice the Gaussian or Normal distribution, and extract the parameters that most successfully define that distribution.

4.2.1. Density Estimation

Density estimation problem can be formally defined as follows [46]: given a set of *N* points in *D* dimensions, $x_1, x_2, ..., x_N \in \Re^D$, and a family F of probability density functions on $\Re^D$, find the probability density $p(x) \in F$ that is most likely to have generated the given points. We can define the family F by giving each of its members the same mathematical form, but we should use a set of parameters θ that distinguishes different members by different values (namely mixing probabilities, means and standard deviations)[6].

$$\theta = (\theta_1, ..., \theta_K) = \{(\lambda_1, \mu_1, \sigma_1), ..., (\lambda_K, \mu_K, \sigma_K)\}$$

Lets define the functions in family F in the following mathematical form:

$$p(x;\theta) = \sum_{k=1}^{K} \lambda_k \cdot N(x, \mu_k, \sigma_k) \qquad 4.1$$

[6] λ_k is called, in some documents, as weighting of the k' th gaussian mixture instead of mixing probability.

where $N(x, \mu_k, \sigma_k)$ represents the normal distribution or Gaussian distribution of x.

$$N(x, \mu_k, \sigma_k) = \frac{1}{\left(\sqrt{2\pi} \cdot \sigma_k\right)^D} e^{-\frac{1}{2}\left(\frac{\|x-\mu_k\|}{\sigma_k}\right)^2} \qquad 4.2$$

Equation 4.2 is D dimensional isotropic (having the same properties in all directions) Gaussian function. Each Gaussian function integrates to one 5.3.

$$\int_{\Re^D} N(x, \mu_k, \sigma_k) dx = 1 \qquad 4.3$$

Since $p(x;\theta)$ is a density function, it must be nonnegative and integrate to one as well.

$$\begin{aligned}&\int_{\Re^D} p(x;\theta)dx = \int_{\Re^D} \sum_{k=1}^{K} \lambda_k N(x, \mu_k, \sigma_k)dx = \\ &\sum_{k=1}^{K} \lambda_k \int_{\Re^D} N(x, \mu_k, \sigma_k)dx = \\ &\sum_{k=1}^{K} \lambda_k = 1\end{aligned} \qquad 4.4$$

Using Gaussian Mixture Models (GMM) is a suitable tool of modeling clusters of points by assigning a Gaussian for each cluster with its mean somewhere in the middle of the cluster, and with a standard deviation that somehow measures the spread of that cluster [46].

Gaussian Mixture Models (GMM) for density estimation is popular for two main reasons [44]:

- They can be reliably computed by the efficient Expectation Maximization (EM) algorithm.
- They provide a generative model for the way the data may have been created.

The last property is the main reason of common use of GMM's for unsupervised clustering because most other clustering algorithms are not generative, and therefore cannot provide predictions regarding previously unseen points. We can express what we mean by generative as the following procedure. Assume that we have a cloud of points such as in *Figure 26*.

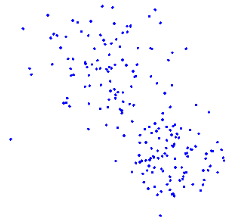


Figure 26: Clouds of data points.

Those data clouds could have been generated by repeating the following procedure *N* times (300 for those particular data points), once for each point x_n:

- Draw a random integer between 1 and *K* with probability λ_k of drawing *k*. This selects the cluster from which to draw point x_n.
- Draw a random *D*-dimensional real vector $x_n \in \Re^D$ from the *k*-th Gaussian density $N(x, \mu_k, \sigma_k)$.

This is called a *generative model* for the given set of points [44].

4.2.2. Maximum Likelihood Estimation

Now we have another problem of finding vector θ that is parameter set, which specifies the model from which the points are most likely to be drawn. In order to determine the meaning of "*most likely*" we need a function $L(X;\theta)$ (similar to the distribution function p) that measures the likelihood of a particular model given the set of points X . A general description of likelihood function is

$$L(X;\theta) = \prod_{n=1}^{N} p(x_n;\theta) \qquad 4.5$$

If we talk about the mixture of models then the equation 4.5 becomes

$$L(X;\theta) = \prod_{n=1}^{N} \sum_{k=1}^{K} \lambda_k N(x_n, \mu_k, \sigma_k) \qquad 4.6$$

In a more precise way, we can determine the parameter set by means of the likelihood function for a Gaussian Mixture Model as follows.

$$\hat{\theta} = \arg\max{}_{\theta} L(X;\theta) \qquad 4.7$$

4.2.3. Characterization of Likelihood

The logarithm of the likelihood function in 4.6 is

$$LL(X;\theta) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \lambda_k N(x_n, \mu_k, \sigma_k) \qquad 4.8$$

In order to make best estimates for parameter set, which gives maximum likelihood, we should compute the derivatives of $LL(X;\theta)$ with respect to $\lambda_k, \mu_k, \sigma_k$ [47].

$$\frac{\partial LL}{\partial \mu_k} = \frac{\partial LL}{\partial \lambda_k} = \frac{\partial LL}{\partial \sigma_k} = 0 \qquad 4.9$$

The derivative functions we had above are non-linear and non-analytically solvable equations but there is a much faster and recently popular algorithm, Expectation and Maximization [44,46].

EM algorithms may be described briefly as follows. Detailed information can be found in Moore's tutorial notes[7] and [46]. Given an initial estimates $[\lambda_k, \mu_k, \sigma_k]^{(0)}$ repeat the following steps until the convergence to a local maximum of the likelihood function:

E – Step: $$p^{(i)}(k \mid n) = \frac{\lambda_k^{(i)} N(x_n, \mu_k^{(i)}, \sigma_k^{(i)})}{\sum_{k=1}^{K} \lambda_k^{(i)} N(x_n, \mu_k^{(i)}, \sigma_k^{(i)})}$$

M – Step: $$\mu_k^{(i+1)} = \frac{\sum_{n=1}^{N} p^{(i)}(k \mid n) x_n}{\sum_{n=1}^{N} p^{(i)}(k \mid n)}$$

$$\sigma_k^{(i+1)} = \sqrt{\frac{1}{D} \frac{\sum_{n=1}^{N} p^{(i)}(k \mid n) \cdot \left\| x_n - \mu_k^{(i+1)} \right\|^2}{\sum_{n=1}^{N} p^{(i)}(k \mid n)}}$$

$$\lambda_k^{(i+1)} = \frac{1}{N} \sum_{n=1}^{N} p^{(i)}(k \mid n)$$

4.3. Artificial Neural Networks

Neural network technology can be considered as the fashionable research and development area in Artificial Intelligence. The Neural Network technology in computer science is in fact inspired of human nervous system and the Neural Network term is especially a biological term itself. In computer science the Artificial Neural Network term may be considered as the matching definition of the biological NN.

[7] Some documents of Andrew W. Moore about Statistical Data Mining Tutorials and Artificial Intelligence notes were used which may be found online at http://www-2.cs.cmu.edu/~awm/tutorials/

A real neural network is a composition of a few (about 100) billion tiny cells those called neurons connected to thousands of other neurons and communicate with them via electrochemical signals in our brain. Also artificial neural network tends to model those biological structure and architecture.

If we compare the human nervous system and a serial computer with very high speed, a serial computer requires millions of computations to perform even the simplest functions in human life such as driving a car. Although serial computer technology we have has the capability of making billions of computations per second, the processing needed would have to be at least as fast as today's fastest computers if we had already implemented such a complex activity. In fact the way that brain processes information is very different from a serial computer. Although biological neurons are relatively slow in transmitting information, we are somehow able to complete the information process that it takes to drive a car. This is because our nervous system does not process serially. Instead, the human brain is an example of a *massively – parallel system* rather than executing computations one-by-one in series, the human brain makes many, many computations simultaneously when performing a neural process [37].

There are some comparisons that Bullinaria [4] has made between human nervous system and parallel computer technology.

1. There are approximately 10 billion neurons in the human cortex, compared with 10 of thousands of processors in the most powerful parallel computers.
2. Each biological neuron is connected to several thousands of other neurons, similar to the connectivity in powerful parallel computers.
3. Lack of processing units can be compensated by speed. The typical operating speed of biological neurons is measured in milliseconds (10^{-3} s), while a silicon chip can operate in nanoseconds (10^{-9} s).

4. The human brain is extremely energy efficient, using approximately 10^{-16} joules per operation per second, whereas the best computers today use around 10^{-6} joules per operation per second.
5. Brains have been evolving for tens of millions of years; computers have been evolving for tens of decades.

4.3.1. The Neuron

Basically the neurons in human nervous system encode their activations or outputs as a series of electrical pulses and consist of synapses, the soma (cell body), the axon, and dendrites. The *soma* processes the incoming activations and converts them into output activations. *Dendrites* are fibres, which emanate from the cell body and provide the receptive zones that receive activation from other neurons. *Axons* are fibres acting as transmission lines that send activation to other neurons. The junctions that allow signal transmission between the axons and dendrites are called *synapses*. The process of transmission across the synaptic cleft is generated by diffusion of chemicals called *neurotransmitters* [4,31].

Operation principle of a real neuron may be summarized as follows. The neuron continuously receives signals from inputs and then performs a summation of inputs to itself in some way and then, according to the result of a non – linear activation function (i.e. some threshold value), the neuron arouses, in the other words it generates a voltage and outputs a signal along axon. This simplified model is known in literature as Threshold Logic Unit (a.k.a. McCulloch-Pitts Neuron, MPN).

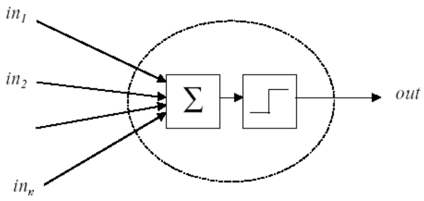
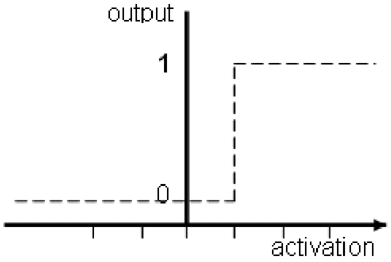
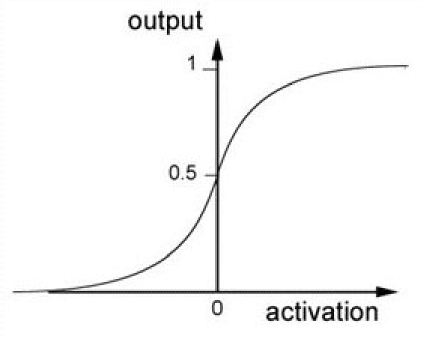


Figure 27: Simple McCulloch-Pitts Neuron



$$\operatorname{sgn}(x)=\begin{cases}1 & \text{if x}\geq 0\\ 0 & \text{if x}<0\end{cases} \tag{4.10}$$



$$\operatorname{Sigmoid}(x)=\frac{1}{1+e^{-x}} \tag{4.11}$$

Assume that the corresponding output function $f(x)$ of input x is defined as (according to the equation 4.10)

$$f(x)=\operatorname{sgn}(\sum\nolimits_{i=1}^{N} x_i-\theta) \tag{4.12}$$

where θ is the threshold value. Then the output is

$$f(x)=\left\{\begin{matrix} 0 & \sum_{i=1}^{N} x_i < \theta \\ 1 & \sum_{i=1}^{N} x_i \geq \theta \end{matrix}\right\} \qquad 4.13$$

But we should care that MP Neuron is the very simplified model of real neuron. If we extend this model, we come up with an extended neuron type, which is the basis element of feed forward network.

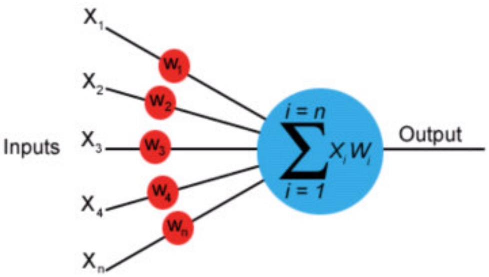


Figure 28: Weighted input neuron. Each input has its own activation value called weightings.

One or two neurons are not enough in practice so we have to construct a network consists of many neurons connected each other. The basic component of this structure is called perceptron, which is an arrangement of one input layer of MP neurons feeding forward to one output layer of MP neurons.

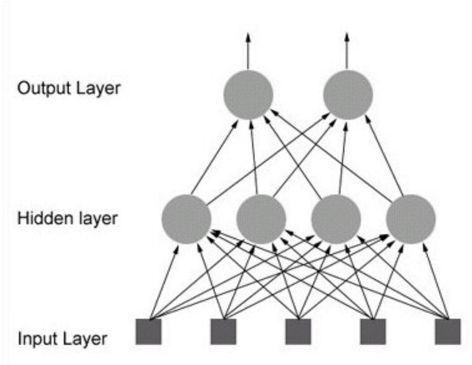


Figure 29: Basic perceptron schema and a complete ANN that consists of perceptrons[8].

4.3.2. Architecture and Topology

There are three common ANN architecture exist in literature. Those are:

1. Single-Layer Feedforward NN: Consists of one input layer to one output layer. No feedback connections.
2. Multi-Layer Feedforward NN: Similar to the single layer NN except there are one or more hidden layers of processing units between input and output layers. No feedback connections are available as well.
3. Recurrent NN: Any network of above two kinds with at least one feedback connection. It may, or may not, have hidden units.

We can represent the general ANN architecture as *weighted directed graphs* mathematically. Below the figure those three kinds were illustrated.

[8] Figure was taken from and http://www.ai-junkie.com

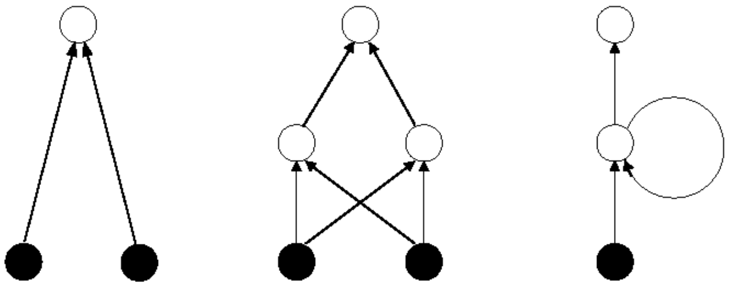


Figure 30: Simple Perceptron, Multilayer Perceptron and Simple Recurrent Network [4]

4.3.3. Learning and Generalization in Neural Nets

As we have mentioned before, there are two specific elements in the basis of an ANN. Weights and thresholds. Depending on the architecture and the topology of an ANN, there are different kinds of learning methods that consist of different adjustment methods setting those two elements.

If we have two inputs, then the weights define a decision boundary that is a one-dimensional straight line in the two dimensional *input space* of possible input values. If we have *n* inputs, the weights define a decision boundary that is a "*n-1*" dimensional *hyperplane* in the *n* dimensional input space. This hyperplane is clearly linear and we can only divide the space into two regions. Problems with input patterns, which can be classified using a single hyperplane, are said to be *linearly separable* [4]. Generally, we have to deal with input patterns that are not binary, and also we expect from neural networks to form complex decision boundaries among the classes. Problems (such as XOR), which cannot be classified in this way are said to be *non-linearly separable.*

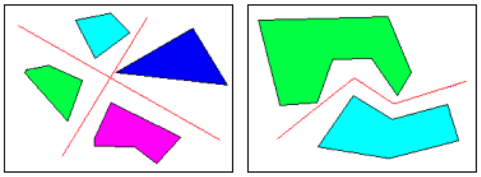


Figure 31: Linear and non – linear separable classes. The axes represent input patterns (two dimensional).

Whether we have a simple perceptron or a complex multilayer NN, we need to develop a procedure for determining correct weightings. Direct computation of the weightings is almost impossible so we have to use a different schema e.g. start with random initial weightings and adjust all the values in small steps until reaching a desired value iteratively. It can be shown that [4] if a problem is linearly separable, then this rule (perceptron learning rule) will find a set of weights in a finite number of iterations that solves the problem correctly.

Some of those more popular training methods include back – propagation, the delta rule, and Kohonen learning.

Most of the learning rules can be categorized into two areas:

1. *Supervised learning rules:* The learning algorithms in this category are not able to decide the output value of a given input pattern themselves. So an external interference is needed to make the decision of the output. This kind of algorithms adjusts all the necessary weightings according to the given input – output pair. This may be very complicated in some networks. Supervised networks include back-propagation and the delta rule.
2. *Unsupervised learning rules:* As having difference from the supervised learning rules, this kind of learning algorithms are able to evaluate the corresponding output with a given input pattern without any external

intervention. Kohonen learning algorithm is the most popular one in that kind.

After training, the network must also be able to generalize, correctly classify *test patterns* it has never seen before. NN's make generalization according to the training data as well. If the training data contains errors, probably it does, the generalization would be worse. If we recover a function approximation on training data in the case of we only have noisy data points, we can expect the NN give better generalization results regardless of the noisy training data points [4].

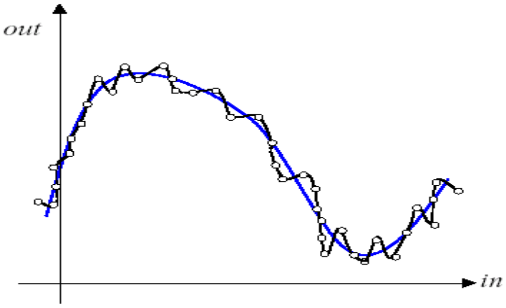


Figure 32: Function approximation.

4.3.3.1. Supervised Learning Algorithms

4.3.3.1.1. Perceptron Learning Rule

Assume that the above unit j, we have the parameters objective output as $\mathrm{targ}_{\mathrm{j}}$ and current output as $out_j = \mathrm{sgn}\left(\sum_{i=1}^{N} in_i \cdot w_{ij}\right)$ where in_i is the activation of the previous layer neurons (previous layer has N neurons). If we check out all the possibilities of target and current output values (in this case the values are distinct 0 or 1 because of the activation function “sgn”), we can generalize the weighting adjustment as the following [4].

$$\Delta w_{ij} = \eta \, (targ_j - out_j) \, in_i \qquad 4.14$$

This weight update equation is then called Perceptron Learning Rule where the parameter η is called as *learning rate* or *step size*. The weight adjustment should be applied repeatedly every weighting in network until the training process converges to a solution [4].

4.3.3.1.2. Error Minimization

The Perceptron Learning Rule is an algorithm for adjusting the network weights minimizes the difference between the actual outputs and the desired outputs. We can define a sum squared error function to quantify this difference. It is the total squared error summed over all output units *j* and all training patterns *p*.

$$E(w_{ij}) = \tfrac{1}{2}\sum_{p}\sum_{j}\left(targ_j - out_j\right)^2 \qquad 4.15$$

The aim of learning is to minimize this error by adjusting all the weights. A systematic procedure for doing this requires the knowledge of how the error $E(w_{ij})$ varies as we change the weights, i.e. the gradient of *E* w.r.t w_{ij} .

Suppose that we have a function $f(x)$ changing x by an amount.

$$\Delta x = x_{new} - x_{old} = -\eta \frac{\partial f}{\partial x} \qquad 4.16$$

where η is a small nonnegative constant specifying the amount changing x by and the derivative determines the direction to go in. If we repeatedly use this equation then the function keeps descending to its minimum [4]. If we modify the equation 4.16 for weighting variables so we have the following.

$$\Delta w_{kl} = -\eta \frac{\partial E(w_{ij})}{\partial w_{kl}} \qquad 4.17$$

4.3.3.1.3. Back Propagation Algorithm

All the equations above are the subject of Single Layer Perceptron. If we take a look at the situation in Multi Layer Perceptron, first we should define the form of network.

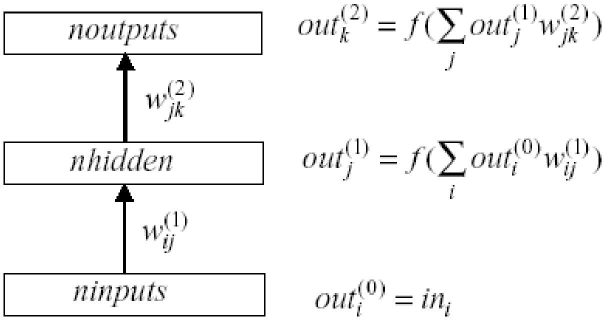

noutputs
$w_{jk}^{(2)}$
nhidden
$w_{ij}^{(1)}$
ninputs
$out_k^{(2)} = f(\sum_j out_j^{(1)} w_{jk}^{(2)})$
$out_j^{(1)} = f(\sum_i out_i^{(0)} w_{ij}^{(1)})$
$out_i^{(0)} = in_i$

Figure 33: Two layered Multi – Layer Perceptron

Assume that we want to adjust the weighting value $w_{ij}^{(n)}$ in order to minimize the error function. The equation 4.15 becomes

$$E(w_{ij}^{(n)}) = \tfrac{1}{2}\sum_{p}\sum_{j}\left(targ_j^p - out_j^{(N)}(in_i^p)\right)^2 \qquad 4.18$$

Where N represents the last layer of net. Also the form of gradient descending weight updates in equation 4.17 becomes

$$\Delta w_{kl}^{(m)} = -\eta \frac{\partial E(w_{ij}^{(n)})}{\partial w_{kl}^{(m)}} \qquad 4.19$$

If we compute the partial derivatives and map it to the weight update function then we have

$$\Delta w_{hl}^{(n)} = \eta \sum_{p} delta_l^{(n)} . out_h^{(n-1)} \qquad 4.20$$

Where deltas can be computed as

$$delta_k^{(N)} = \left(targ_k - out_k^{(N)}\right).f'\left(\sum_{j} out_j^{(1)} w_{jk}^{(N)}\right) = \left(targ_k - out_k^{(N)}\right).out_k^{(N)}.\left(1 - out_k^{(N)}\right) \qquad 4.21$$

where N is the last (output) layer. The delta values of the other layers may be computed as follows.

$$delta_k^{(n)} = \left(\sum_{k} delta_k^{(n+1)}.w_{lk}^{(n+1)}\right).f'\left(\sum_{j} out_j^{(n-1)} w_{jk}^{(n)}\right) = \left(\sum_{k} delta_k^{(n+1)}.w_{lk}^{(n+1)}\right).out_k^{(n)}.\left(1 - out_k^{(n)}\right) \qquad 4.22$$

We can smooth out the weighting updates, without slowing down the learning too much, by updating the weights with the moving average of the individual weight changes corresponding to single training patterns. The equation is an extended version of equation 4.22. There is an additional part containing *momentum* parameter α having a multiplier of previous weighting change. Momentum parameter α may be set between 0 – 1. Good sizes of α depend on the size of the training data set and how variable it is [4].

$$\Delta w_{hl}^{(n)}(t) = \eta \sum_{p} delta_{l}^{(n)}(t).out_{h}^{(n-1)}(t) + \alpha.\Delta w_{hl}^{(n)}(t-1) \qquad 4.23$$

4.3.3.2. Unsupervised Learning Algorithms

Unsupervised learning is a generic term that is including some of pattern recognition algorithms such as vector quantization (more generally clustering), Gaussian Mixture Models, or some kind of neural networks (Kohonen SOM). In the neural network concept, I think it will be better to use term Competitive Learning instead. Motivation of the competitive learning is the competition (instead of output feeding) of the output neurons for a given input pattern without external collimation and cooperation.

If we take a look at from the neural network side, competitive learning rules are based on some objectives from neurons activity and the total behaviour of the network. First of all, the network should automatically categorize prominent features of training data set. Another expectation is that the network should be capable of finding clusters in training data set that can be used for generalization. Shortly it is assumed that the input vectors share common features and the network is able to identify those features. Basic stages of competitive learning can be lined up as following.

Competition: Since the neurons are forced to organize themselves, given an input pattern, outputs compete to see who is winner based on a discriminator function, which provides basis for competition and may be selected as the distance function between the input vector and weight vector. The winning neuron is called as WTA (winner takes all neuron).

Cooperation: A topological neighborhood area, which contains the affected neurons within, is selected centered on location of the winning neuron.

Synaptic Adaptation: Adjust weightings of the neurons, which belong to the neighborhood area of the winning neuron so that cluster boundaries become more distinct. This leads network presenting a similar input would result in enhanced response from winner, which means generalization.

As it can be understand from the cooperation and adaptation phases, not only the winning neuron is modified but its neighbors as well. This allows considering the SOM to be a nonlinear PCA projecting data onto a lower-dimensional display [4].

4.3.3.2.1. Kohonen Self Organizing Maps

Kohonen SOM is the most common unsupervised neural network algorithm, which is expressed by its inventor's name, Teuvo Kohonen [26].

Kohonen network is one or two-dimensional discrete topographic map of input pattern vectors of N dimension. In the architecture of the SOM, there are one input layer and one computational layer, which is one or two-dimensional lattice. Each neuron of the input layer is fully connected to the computational layer. The aim of this structure can be expressed as an imitation that different sensory inputs such as visual or auditory inputs are mapped onto corresponding areas of the cerebral cortex.

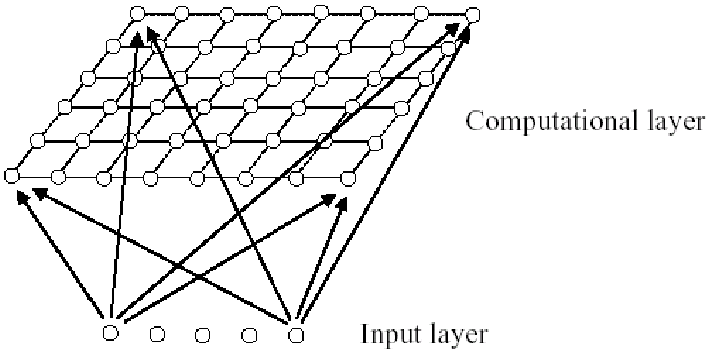


Figure 34: Typical SOM architecture, from Bullinaria [4].

Assume that we have training set of D dimensional input vectors. So the neuron count in input layer will be D then $x = \{x_i : i = 1,..,D\}$. Lets set the dimension of computational layer as $N = R \times C$ where N is the number of neurons in that layer so the corresponding weights are $w = \{w_{ji} : j = 1,..,N; i = 1,..,D\}$.

For competition phase, we have to define a discriminator function such as squared Euclidian distance. In order to determine the winning neuron we compute the distance of input vector – corresponding weightings for each neuron j and declare the winning neuron, which minimizes the distance.

$$d_j(\mathbf{x}) = \sum_{i=1}^{D} (x_i - w_{ji})^2 \qquad 4.24$$

It is observed in neurobiology that there is a lateral interaction between the neighboring excited neurons. But this neighborhood declines with distance, which means that closest neurons to the stimulated neuron tend to get excited more than the others.

If we look generally to the training phase, after identification of the winning neuron, cooperative stage, which acts as lateral interaction between the neighbors of the winning neuron. In order to state the excitation level of the neighboring neurons, a neighbor function should be defined, which shrinks the area of the neighborhood over time. Some formulations of the neighborhood function can be found in literature e.g. Gaussian function of radius or exponential decay function, which diminishes over time. Iteration of this process over the time makes the regions on topographic map distinct and clear.

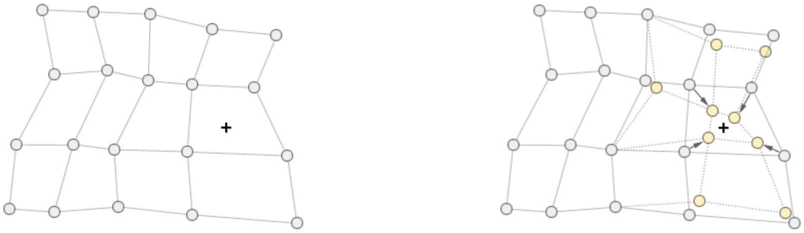


Figure 35: Adaptation of winning neuron and its neighbors. Cross represents input vector coordinates.

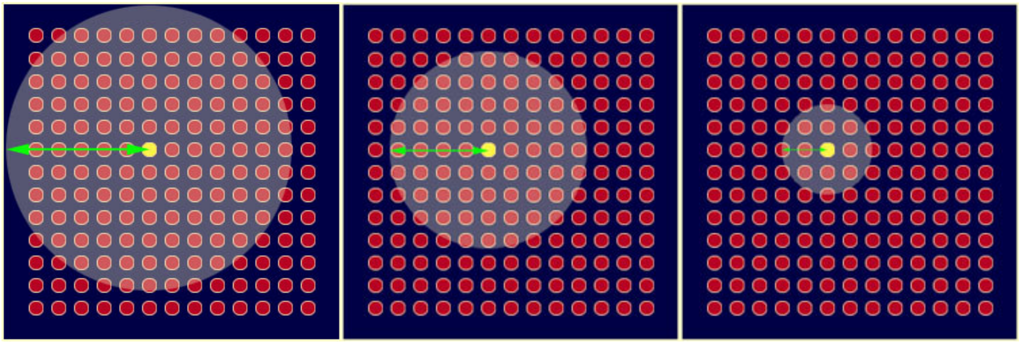


Figure 36: Decay of the neighboring function with respect to time[9].

[9] Figure was taken from http://www.ai-junkie.com/

After cooperation phase we have mentioned above, the network should adjust the weights of winning neuron as an *adaptive learning process* and all of the neurons comprised by neighborhood according to the following notation.

$$\Delta w_{ji} = \eta(t) \cdot T_{j,I(x)}(t).(x_i - w_{ji}) \qquad 4.25$$

If we scrutinize the equation above, weight adjustment of each neuron depends on the time dependent learning rate and topological neighborhood effect. If we select the equation 4.26 as a popular exponential decay function in which the size of neighborhood size σ decreases in time, we can easily define topological neighborhood affiliated with time as in 4.27.

$$\sigma(t) = \sigma_0 \exp(-t / \tau_\sigma) \qquad 4.26$$

$$T_{j,I(x)} = \exp(-S_{j,I(x)}^2 / 2\sigma^2) \qquad 4.27$$

S_{ij} represents the lateral distance between the neurons i and j on the neural grid and *I(x)* provides the index of winning neuron. Those parameters ensure that the topological neighborhood is maximized around the winning neuron and monotonically decreases to zero as the distance goes infinity [4].

Also there is another parameter in equation 4.25, which is time dependent learning rate. In this case the learning rate decreases in time as neighborhood size does.

$$\eta(t) = \eta_0 \exp(-t / \tau_\eta) \qquad 4.28$$

If we present the general organization of SOM in a condensed form, first neuron weight vectors are initialized randomly. This stage is called initialization. After that, from the input space, a training vector is chosen randomly or sequentially. This is called sampling. The winning neuron *I(x)* is assigned by choosing the closest weight vector to the input vector. This stage is called as matching. After winning neuron selection, weight vectors of the winning neuron and the others are adjusted. This is called updating. The final step, continuation, goes throughout a phase during which the feature map is fine-tuned and comes to provide an accurate statistical quantification of the input space [4].

CHAPTER 5

EXPERIMENTAL RESULTS AND DISCUSSIONS

In this chapter, we will investigate the experimental results of the application written by author and discuss about the results.

5.1. Experimental Setup

In order to execute basic operations and tests, complete DSP featured software was designed and written by author. Object pascal was used as programming language and Borland Delphi IDE was used as software development kit. Delphi was chosen since it has vast variety of visual components library, which makes our job easy to recognize although the programming language is really not good at performance issues. The application software contains all DSP libraries, feature extraction and pattern recognition algorithms and mathematical libraries together.

Real acoustic underwater sound recordings have been evaluated in this thesis work and it is avoided using artificially generated data. The sound library consists of thirty different files, which were recorded from different platform types, classes and different vehicles in different conditions.

In order to improve the accuracy of target classification, a taxonomy-based approach was used instead of direct target identification. For example, platform type recognition were tested first such as the sound identifies a surface or subsurface target before identifying the vehicle as MS Korutürk. The taxonomy-based table of the files was listed below.

Table 1: Sonar data files, which has been used for application. The sound data files were recorded from different targets.

File Name	Sampling Rate	Duration	Platform	Type	Class	Condition
UN_B_1	11025	2:37.542	U	U	UN	B
UN_B_2	11025	1:14.312	U	U	UN	B
UN_B_3	11025	1:41.569	U	U	UN	B
UN_B_4	11025	1:16.557	U	U	UN	B
UN_S_1	11025	2:57.012	U	U	UN	Sn
UN_S_2	11025	1:33.377	U	U	UN	Sn
UN_S_4	11025	2:02.967	U	U	UN	Sn
SB_D_1	11025	2 :05.724	S	S	SB	D
SB_D_2	11025	2 :07.121	S	S	SB	D
SB_T_1	11025	1: 30.180	S	S	SB	GT
SB_T_2	11025	1: 37.785	S	S	SB	GT
SB_T_3	11025	0: 28.242	S	S	SB	GT
SB_T_4	11025	0: 28.047	S	S	SB	GT
UI_B_1	11025	2:31.023	U	U	UI	B
UI_B_2	11025	1:26.659	U	U	UI	B
UI_B_3	11025	1:43.318	U	U	UI	B
UI_B_4	11025	1:09.307	U	U	UI	B
UI_S_1	11025	3:50.338	U	U	UI	S
UI_S_2	11025	1:07.462	U	U	UI	S
UI_S_3	11025	0:33.274	U	U	UI	S
UI_S_4	11025	0: 14.238	U	U	UI	S
G_1	11025	0: 17.818	S	S	G	D
G_2	11025	1:16.625	S	S	G	D
G_3	11025	0:45.233	S	S	G	D
T_1	11025	0:56.919	S	S	T	D
T_2	11025	1:18.665	S	S	T	D
Y_1	11025	0: 44.962	S	S	Y	D
Y_2	11025	0:29.049	S	S	Y	D
Y_3	11025	2:40.802	S	S	Y	D
Y_4	11025	1: 27.506	S	S	Y	D

5.2. Experimental Results

Experiments were based on a basic template that has been applied to the sound data we had. This basic template consists of three abstract processes, Feature Extraction, Target Classification and Target Identification.

5.2.1. Feature Extraction

Feature extraction is the first step of the progression. Three different feature extraction schemas were applied to the sound data. In order to provide consistency of the recognition processes, some feature extraction parameters were kept same for all profiles such as segment length per data vector. Segment lengths per data vector were set to 16384 for all profiles because it was considered that for an acoustic data in which the sample rate is 11025, that data count represents 1.486-second length acoustic information would be enough to extract salient features. One may think that this length is too much for each data vector but in this experiments, it was intended that each data vector should contain as much information as possible such as very low frequency components of target acoustic signatures. Also this let us prevent using silence parts of data as an individual data vector.

It was intended to compare time-frequency domain representation of sound data and measure the effectiveness of usual speech feature extraction schemas such as LPCC and MFCC. Also the schemas were prepared to compare the combinations.

Profile I is basically time domain representation (also scale) of sound data and consists of LPC coefficients of wavelet packages as combinatorial structure and MFC coefficients of input data. In this template, wave data were split into data vectors (up to 100) each consists of 16384 data. Wavelet decomposition schemas that driven by Daubechies filter with order 8 and level 10 were applied to each data vector. 10 LPC coefficients were computed from each vector of decomposition level. Finally we had a matrix, for each data vector; consists of 10 rows and 10 columns.

The first 10 feature data were computed by using mean values of each column and the other 10 feature data were computed by using standard deviation of each column. In the Mel Frequency Cepstral Coefficients computation schema, 12 of 24 channels were used. Hanning window was used as FFT window type and magnitude spectrum was used instead of power spectrum. These MFC coefficients completed the last 12 feature vectors so we had a feature vector that has 32 feature parameters for each data vector. In *Figure 37*, the schematic diagram of Profile I has been implemented.

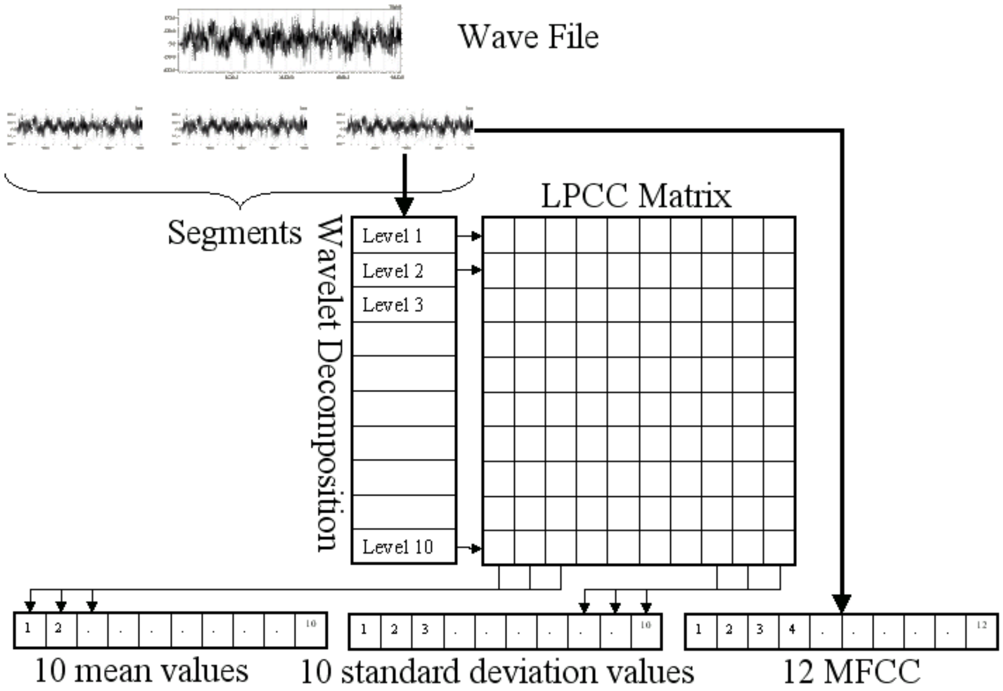


Figure 37: Feature Extraction Schema of Profile I

Profile II consists of only 10 LPC Coefficients plus 12 MFC Coefficients for each data vector and Profile III consists of only 12 MFC Coefficients for each, similarly.

Profile II and III were designed to measure that whether a speech specific feature extraction method is enough for underwater acoustic signal classification or not.

For each of the profiles, training data sets and candidate test sets were generated by application software.

5.2.2. Classification and Identification

For classification and identification processes, a Back Propagation Neural Network, Kohonen Self Organizing Map and Gaussian Mixture Model kernels were implemented. Thus we had a chance of testing both supervised and unsupervised classification algorithms on the same data. Classification and identification processes were based on a taxonomical framework we have mentioned above. Platform, type, class and condition specific classification and identification were studied.

In order to achieve that taxonomy, every training file (pattern of feature vectors or codebooks) should be labeled four times separately.

Identification phase may be classified in two parts such as open-set and closed-set identification. In the latter one, the system is forced to make a decision by choosing the best matching identity in database, no matter what the error rate is. But as a challenging problem, open-set identification should use a threshold level in order to make a decision that unknown patterns point out the best matching identity or not. Open-set identification is used if there is a possibility that unknown patterns are none of the registered identities.

In our case, we can use different identification tasks on different taxonomy levels, in “platform” branch; there is no more possibility of identity than the target is submarine or surface. Closed-set identification may be suitable for that branch. However in “class” branch, the target may belong to any kind of class that is not registered yet, moreover it may be belong to a class we have never seen before.

Open-set identification is suitable for that branch as well. The identification task type decision is not in the concept of this study, so it was left for future investigation to select the best identification type for vast variety of purposes.

The following table shows the common properties of all tests that were made for classification and identification methods. In some of the following test tables, only the test number will be used for simplicity.

Table 2: Common properties of test patterns.

TEST NO	FILE NAME	VECTOR COUNT	SAMPLE LENGTH
1	UN_B_T1	68	16384
2	UN_B_T2	51	16384
3	UN_S_T1	100	16384
4	SB_D_T1	85	16384
5	SB_T_T1	60	16384
6	SB_T_T2	18	16384
7	UI_B_T1	58	16384
8	UI_B_T2	46	16384
9	UI_S_T1	45	16384
10	UI_S_T2	9	16384
11	Y_T1	19	16384
12	Y_T2	58	16384
13	G_T1	11	16384
14	T_T1	52	16384

5.2.2.1. Back Propagation Neural Network

Three-layered neural network has been implemented which has one input, one hidden and one output layer. The number of the neurons of the input layer is equal to feature vector data count. Also hidden layer neuron count is adjustable for the sake of fine-tuning of classification rate. In the experiments, the hidden layer has fifteen neurons and the output layer has different number of neurons since the string sequence of the output neurons represents the index of the identity in small database. This depends on the total identities of input domain. In our experiments, for platform specific tests, we have two output neurons, one represents “S” and the other represents “U”. If we had eight kinds of identity, we should have eight output neurons. In the implemented version of the core neural network, input and output neuron counts were defined dynamically according to the training data labels and taxonomy index.

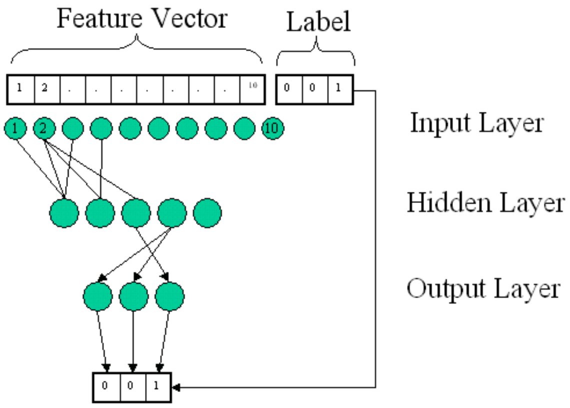


Figure 38: Core neural network

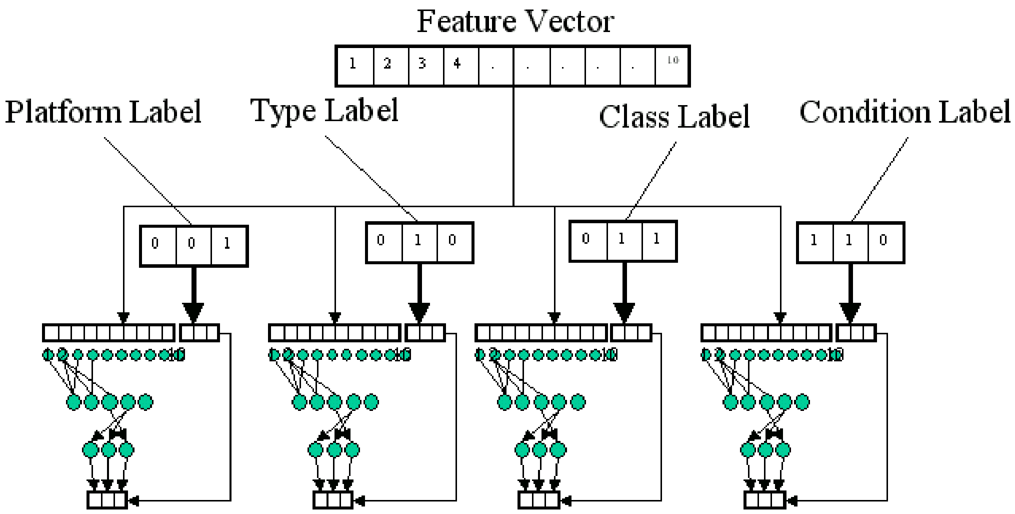


Figure 39: BPNN set comprises four neural network. Each network uses same feature vector but different output value related to the taxonomy index.

In the experiments, we have the following identity counts for each taxonomy levels; two identities for platform, two identities for type, six identities for class and four identities for condition. At the initial state, we adjusted the neural network cores as follows.

Table 3: Neural Network Core initial settings.

Hidden Layer Neurons	15
Alpha Value	1
Momentum	0,9
Epoch per Identity	1000
Train Data Count	1055
Recognition Threshold	0,9

In our network model, we make decisions by examining the output neuron that has the highest value. Assume that we have two output neurons that have the value 0,025 and 0,78 after test pattern. Matching the highest value leads us select the second index as winner, which has 0,22 (1-0,78) error rate. In closed-set identification task, choosing the winner, whatever the value it has, accomplishes the identification task. If we use open-set identification task, a predefined tolerance level should be selected, which is called in our examples as recognition threshold value. If this tolerance level is above 0,22 then we announce that the winner is the second one else we declare that we don't have such identity in our database.

In this phase, one may ask that what the suitable threshold value should be. Of course this value affects the recognition sensitivity and can be adjusted according to the system requirements. But we should have lots of test patterns with different known data but now it is almost impossible to collect that amount of data. In order to provide forethought, we made the following tests on recognition threshold value.

Table 4: Threshold test of two samples. Condition specific matching has been used

	(File No 5)			**(File No 1)**		
Threshold	Correct Match %	Mean Error	Not Classified	Correct Match %	Mean Error	Not Classified
0,50	88	0,0456	5	82	0,0298	3
0,55	88	0,0456	5	81	0,022	5
0,60	88	0,0456	5	81	0,022	5
0,65	88	0,0456	5	81	0,022	5
0,70	88	0,0456	5	78	0,0115	7
0,75	88	0,0456	5	77	0,0078	9
0,80	85	0,0394	8	77	0,0078	9
0,85	75	0,0192	18	77	0,0078	10
0,90	65	0,02	30	74	0,003	13
0,95	65	0,02	32	74	0,003	15

Graph in *Figure 40* is the graphical representation of *Table 4*. As it can be seen from the graph, increasing the value causes mean error rate to decrease. Also this decreases the correct classification rate as well and makes the knowledge of the network smaller for open-set identification tasks.

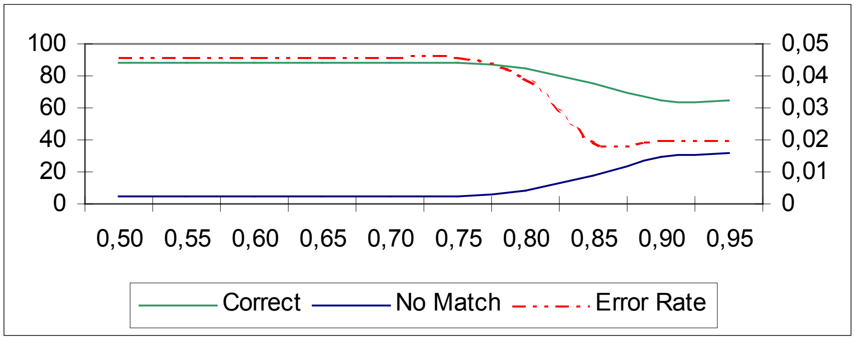


Figure 40: Graphical representation of (File No 5) Table 4.

Table 5: Profile matching results of BPNN, which tested for three profiles. Platform branch of taxonomy has been tested.

	PROFILE I				PROFILE II				PROFILE III			
	Match	Close Match	Wrong Match	No Match	Match	Close Match	Wrong Match	No Match	Match	Close Match	Wrong Match	No Match
1	100	0	0	0	100	0	0	0	99	0	1	0
2	100	0	0	0	100	0	0	0	96	0	0	4
3	100	0	0	0	100	0	0	0	99	0	0	1
4	100	0	0	0	100	0	0	0	100	0	0	0
5	100	0	0	0	100	0	0	0	37	0	40	23
6	100	0	0	0	**X**	**X**	**X**	**X**	0	0	83	17
7	100	0	0	0	100	0	0	0	45	0	22	33
8	100	0	0	0	100	0	0	0	98	0	0	2
9	100	0	0	0	87	0	11	2	87	0	6	7
10	100	0	0	0	0	0	78	22	100	0	0	0
11	**X**	**X**	**X**	**X**	95	0	0	5	42	0	47	11
12	100	0	0	0	100	0	0	0	97	0	0	3
13	55	0	27	18	100	0	0	0	82	0	9	9
14	98	0	0	2	88	0	6	6	40	0	50	10

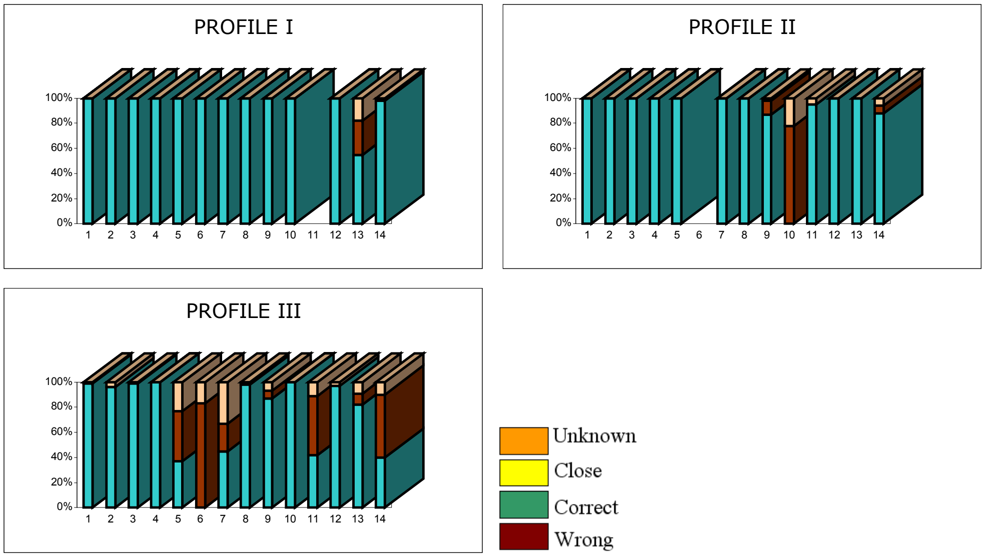


Figure 41: Graphical representation of Table 5

Table 6: Profile matching results of BPNN, which tested for three profiles. Type branch of taxonomy has been tested

	PROFILE I				**PROFILE II**				**PROFILE III**			
	Match	Close Match	Wrong Match	No Match	Match	Close Match	Wrong Match	No Match	Match	Close Match	Wrong Match	No Match
1	100	0	0	0	100	0	0	0	94	0	3	3
2	100	0	0	0	100	0	0	0	98	0	2	0
3	100	0	0	0	100	0	0	0	99	0	0	1
4	100	0	0	0	100	0	0	0	100	0	0	0
5	100	0	0	0	100	0	0	0	48	0	37	15
6	100	0	0	0	**X**	**X**	**X**	**X**	0	0	94	6
7	100	0	0	0	100	0	0	0	66	0	24	10
8	100	0	0	0	100	0	0	0	100	0	0	0
9	100	0	0	0	84	0	9	7	80	0	13	7
10	100	0	0	0	0	0	78	22	89	0	0	11
11	**X**	**X**	**X**	**X**	100	0	0	0	58	0	16	26
12	100	0	0	0	100	0	0	0	95	0	0	5
13	64	0	27	9	100	0	0	0	91	0	0	9
14	92	0	2	6	92	0	0	8	42	0	44	14

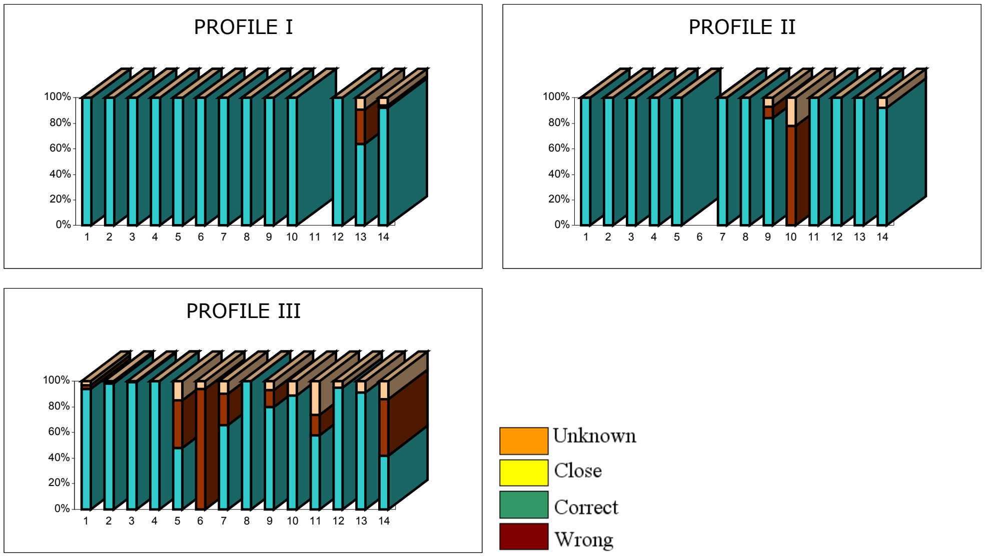


Figure 42: Graphical representation of Table 6.

Table 7: Profile matching results of BPNN, which tested for three profiles. Class branch of taxonomy has been tested

	PROFILE I				PROFILE II				PROFILE III			
	Match	Close Match	Wrong Match	No Match	Match	Close Match	Wrong Match	No Match	Match	Close Match	Wrong Match	No Match
1	75	19	0	6	69	16	0	25	84	3	0	13
2	88	10	0	2	98	0	0	2	92	2	2	4
3	100	0	0	0	97	0	0	3	98	0	0	2
4	100	0	0	0	100	0	0	0	100	0	0	0
5	58	2	28	12	60	2	0	38	57	0	26	17
6	100	0	0	0	**X**	**X**	**X**	**X**	83	0	0	17
7	100	0	0	0	100	0	0	0	85	0	7	8
8	22	67	0	11	100	0	0	0	96	4	0	0
9	100	0	0	0	84	0	7	9	44	4	21	31
10	100	0	0	0	34	0	44	22	100	0	0	0
11	**X**	**X**	**X**	**X**	68	0	0	32	5	0	74	21
12	100	0	0	0	100	0	0	0	100	0	0	0
13	82	0	0	18	82	0	0	18	73	0	9	18
14	44	6	19	30	46	20	0	34	10	0	67	23

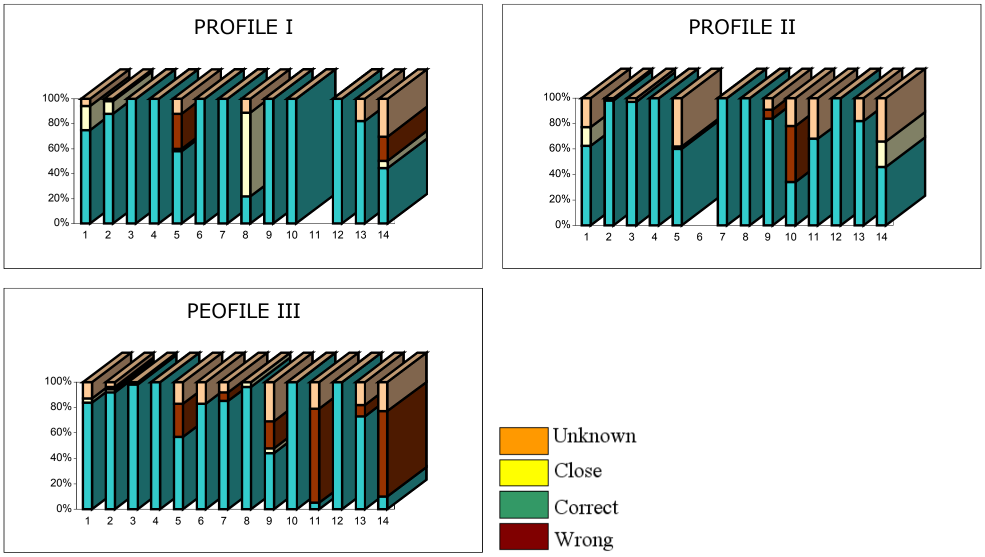


Figure 43: Graphical representation of Table 7.

Table 8: Profile matching results of BPNN, which tested for three profiles. Condition branch of taxonomy has been tested

	PROFILE I				PROFILE II				PROFILE III			
	Match	Close Match	Wrong Match	No Match	Match	Close Match	Wrong Match	No Match	Match	Close Match	Wrong Match	No Match
1	74	13	0	13	0	99	0	1	57	24	0	19
2	8	82	0	10	100	0	0	0	90	8	0	2
3	100	0	0	0	100	0	0	0	100	0	0	0
4	100	0	0	0	100	0	0	0	100	0	0	0
5	65	3	2	30	0	32	0	68	0	0	53	47
6	0	100	0	0	**X**	**X**	**X**	**X**	0	0	33	67
7	93	2	0	5	4	1	0	95	35	0	8	57
8	92	4	0	4	96	0	0	4	96	4	0	0
9	62	20	2	16	0	73	0	27	7	44	25	24
10	56	22	0	22	100	0	0	0	67	0	0	33
11	**X**	**X**	**X**	**X**	100	0	0	0	37	0	16	47
12	100	0	0	0	100	0	0	0	100	0	0	0
13	82	0	18	0	100	0	0	0	100	0	0	0
14	92	0	4	4	85	0	0	15	52	0	17	31

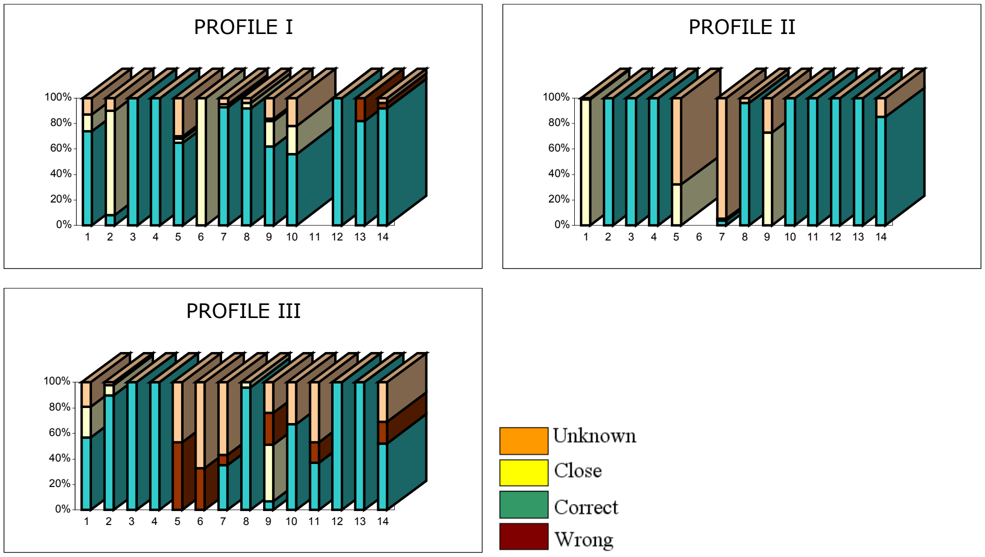


Figure 44: Graphical representation of Table 8.

As it can be seen from *Figures 41-44*, best matching performance classifier can be achieved with the first profile in which we used wavelet decomposition, LPC and MFCC together. The weakness of the other profiles can be considered as a result of less data length of feature vectors and the most important one, MFCC is basically designed for human auditory system. Thus we should avoid using MFCC without collaboration of other feature extraction methods. In profile III, we have used only MFCC based feature vectors and it can be seen how terrible the result is although there are just two identities in that taxonomy branch.

If we remember the basic properties of feature vectors from section three, high between class variation and low within class variation are the most important ones. But in practice, one feature extraction method is not enough to overcome both.

Singularities in test results are negligible, e.g. the thirteenth sample of profile one and tenth sample of profile two. So Profile II should be considered as suitable feature extraction profile also, even it is not as successful as Profile I. The weakness of LPC derived features in Profile II comes from the time variant properties of underwater acoustic channel.

Considerations we have made in “platform” branch are valid for “type” branch, profile one is the best feature extraction method and profile three is the worst one.

In *Figures 43 and 44*, there is an obvious descent of success levels for all profiles concurrently, which is considered as a result of increasing identity variation. The first profile tests are quite satisfactory if we consider the close matching rate in the vicinity of correct matching rate. One may ask that who will decide the network response for unknown pattern is “close to” kind response. In our examples, UI class target tests were resulted as UN. Who will decide that this answer is close-to, if we have no idea about unknown object navigating somewhere? In that case, intelligence reports come to the stage, so the search platform usually has information of expected target identities. If we are expecting UI but we have UN as the response of network, then we evaluate this response as close-to UI and in fact the target is UI.

As overall evaluation of test results, we can say that

- Best performance classifier of BPNN may be achieved with Profile I feature vectors.
- Prosperity levels decreases with increasing variety of identities in relevant taxonomy branches.
- Increasing train patterns provides more stable and correct matching rates.
- Increasing test pattern size reduces mismatches, since feature vector patterns may have common vectors of different objects.

Also condition branch of taxonomy will not be suitable for correct classification. In *Figure 44*, there are untidy ripples of matching rates. Thus we should make some modifications on "condition" branch of taxonomy, or reconstruct taxonomy. Although we have made this important determination, we used this taxonomy for remaining evaluations.

Table 9: Teach error rates of BPNN after train process.

	Platform	**Type**	**Class**	**Condition**
Profile I	0,01124600	0,01004228	2,00990716	0,51586926
Profile II	0,00597537	0,00498728	2,00783796	9,50406033
Profile III	0,03021344	0,04413256	3,02642999	13,53071731

Table 9 shows effects of *data counts* of feature vectors of profiles on network consistency. Profile II has litter error rate than I but this may be disclosed by data length of feature vectors where II has 22 and I has 32 data. Also *Table 9* supports our negatory comments on Profile III although it has just 12 data in a feature vector.

Table 10: BPNN responses for test patterns, which are not registered yet.

Test Num		Sample Count	Not Classified %
1	Cruise ship	18	44
2	Loud boat	80	3,75
3	Pure tone boat	18	83
4	Putt-Putt	18	33
5	Whinning Propeller	16	69

Table 10 shows that BPNN supports open-set identification tasks. Misclassification rates are too high and combining those rates with other data mining approaches, it will be likely to explore new unregistered patterns for network.

5.2.2.2. Self Organizing Map

Basically it is applicable to use feature vectors that represent different identities at one training sequence. Also this is the normal way of using SOMs. In practice, if we train a network with seven different colors, then we have a topological color map in which the same colors grouped into one distinct location.

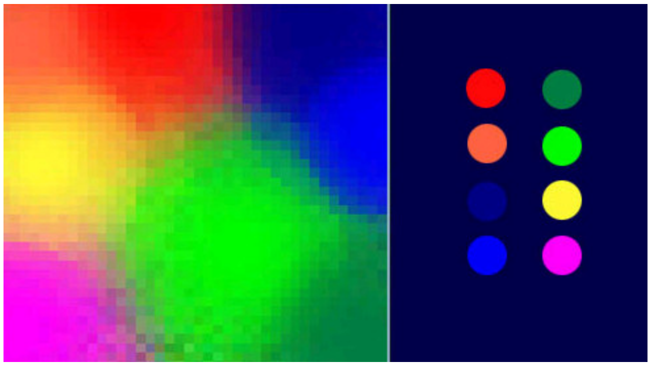


Figure 45: Self Organizing Map trained with 8 different color data.

One may think that if we train SOM with different feature vectors representing different identities may give us reasonable vector quantization or data clusters, e.g. codebook of all database. But in practice increasing the amount of training data and branches of taxonomy tree may cause the map dimension to increase together. As another drawback of this method, increasing the common properties leads a winning neuron to point out different targets at the same time for a test vector. Thus another supplemental approach different from this idea was tested for underwater acoustic features. Instead of using SOM in that way, drawing the topological map of each identity separately and saving those maps represented by the neuron weight vectors labeled with the identity was preferred.

Lets consider only one side of taxonomy, such as class. We have UN, UI, SB, Y, G and T Classes. If we go step by step, first we collect all UN feature vectors from vast vector chunks of training pattern. Train the SOM long time enough for convergence. Save the weight vectors of SOM (the SOM core) to a stream and label this with UN. Repeat those above for other identities. In identification phase, apply the same test pattern for all of the streams, loading registered SOM cores sequentially, and note the minimum distance of each. Finally decide the identity by looking at the minimal errors.

So we have implemented one SOM core and dynamic database of core parameters.

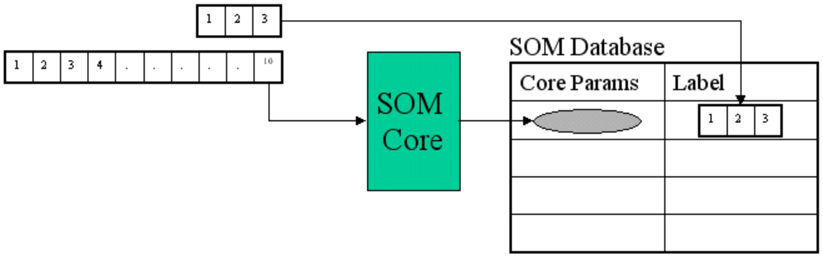


Figure 46: Typical schema of SOM Training process

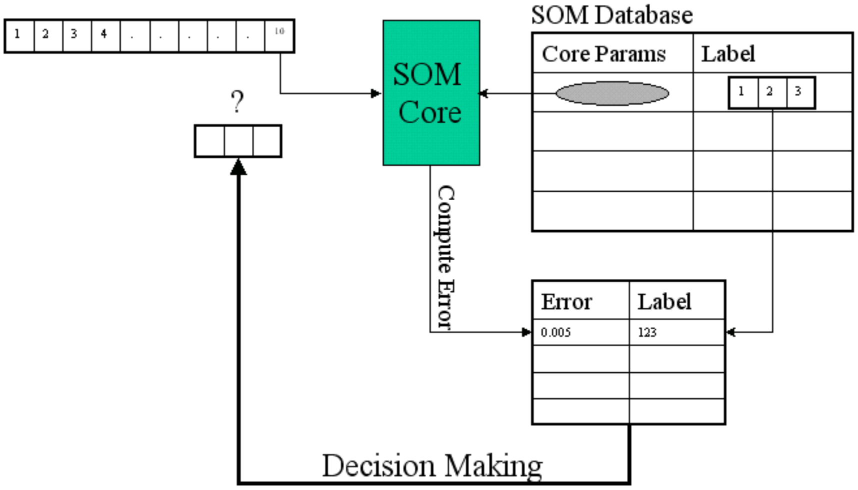


Figure 47: Typical schema of SOM Identification process.

In this type of usage, SOM act as feature extractor for a decision maker. But this is the most beneficial treatment of SOM for acoustic signatures, because completely different labeled feature patterns may contain same acoustic information with targets placed in different branches of taxonomy tree. SB class target in GT condition is definitely “S” platform although the sound information is close to a “U” in “B” condition. Test results exhibited that whatever the epoch count is (it was tested up to 100000 epoch) there were always gaps and intersections between the clusters. Also those test results were presented in this study.

Table 11 exhibits the results of SOM network where SOM act as feature extractor for a decision maker. “C” marks represent correct classification.

Table 11: Correct classification and identification of SOM for the feature vectors extracted by three profile settings. P: Platform, T: Type, Cl: Class, Co: Condition.

	PROFILE I				PROFILE II				PROFILE III			
Test No	P	T	Cl	Co	P	T	Cl	Co	P	T	Cl	Co
1	C	C	C	C	C	C	C	C	C	C	C	C
2	C	C	C	C	C	C	C	C	C	C	C	C
3	C	C	C	C	X	X	X	X	X	X	X	X
4	C	C	C	X	C	C	C	X	C	C	C	X
5	C	C	C	C	C	C	C	C	C	C	C	C
6	C	C	C	C	C	C	C	C	C	C	C	C
7	C	C	C	C	C	C	C	C	C	C	C	C
8	C	C	C	X	C	C	C	X	C	C	C	X
9	C	C	C	C	C	C	C	C	C	C	C	C
10	C	C	C	C	C	C	C	C	C	C	C	C

It is clear that small test patterns cause misclassification and Profile I feature extraction method is the most eligible one.

For accustomed use of SOM, we have applied all test vectors to the SOM core and set the epoch as much as possible such as 75000. After acquisition of a topological map, each training vector was applied again in order to pick up winning neuron among matured collection of neurons. Thus we have labeled the winning neuron according to the taxonomical position of training vector. The neurons may represent more than one identity e.g. 25% UN and 75% UI.

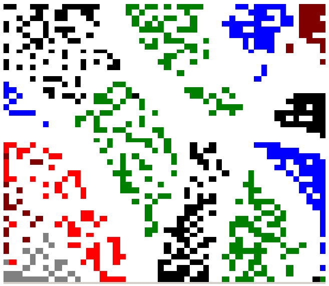


Figure 48: Two (50 X 50) topological maps of a training set that contains 6 identities.

In the figure above, the placements of labeled neurons are the same in both figures since they belong to same train set but represent different identities according to the taxonomy levels. At first sight, SOM generates quite successful codebooks. What about the unlabelled neurons? If we use open-set identification tasks then most of the test patterns will be assessed as "not classified", thus we should use this model for closed-set identification tasks. For this purpose, some experiments have been made with collaboration of Gaussian Mixture Model.

Similar to BPNN and our new generated SOM method, Profile I is the best performance classifier for this model, so only Platform I results have been examined in this study.

Table 12: Test results of customary used SOM with Profile I feature vector settings. Cor: Correct classification rate, Clo: Close classification rate, Wro: Wrong classification rate and Noc: No Classification rate.

Test Num	PLATFORM				TYPE				CLASS				CONDITION			
	Cor %	Clo %	Wro %	Noc %	Cor %	Clo %	Wro %	Noc %	Cor %	Clo %	Wro %	Noc %	Cor %	Clo %	Wro %	Noc %
1	72	0	0	28	72	0	0	28	71	1	0	28	63	9	0	28
2	12	0	0	88	12	0	0	88	12	0	0	88	0	12	0	88
3	99	0	1	0	99	0	1	0	99	0	1	0	99	0	1	0
4	100	0	0	0	100	0	0	0	100	0	0	0	100	0	0	0
5	4	0	0	96	4	0	0	96	4	0	0	96	4	0	0	96
6	55	0	0	45	55	0	0	45	50	5	0	45	0	55	0	45
7	0	0	0	100	0	0	0	100	0	0	0	100	0	0	0	100
8	70	0	0	30	70	0	0	30	22	48	0	30	50	20	0	30
9	2	0	0	98	2	0	0	98	2	0	0	98	0	2	0	98
10	0	0	0	100	0	0	0	100	0	0	0	100	0	0	0	100
11	0	0	95	5	0	0	95	5	0	0	95	5	0	0	95	5
12	97	0	0	3	97	0	0	3	91	0	6	3	97	0	0	3
13	82	0	0	18	82	0	0	18	82	0	0	18	82	0	0	18
14	67	0	0	33	67	0	0	33	19	48	0	33	67	0	0	33

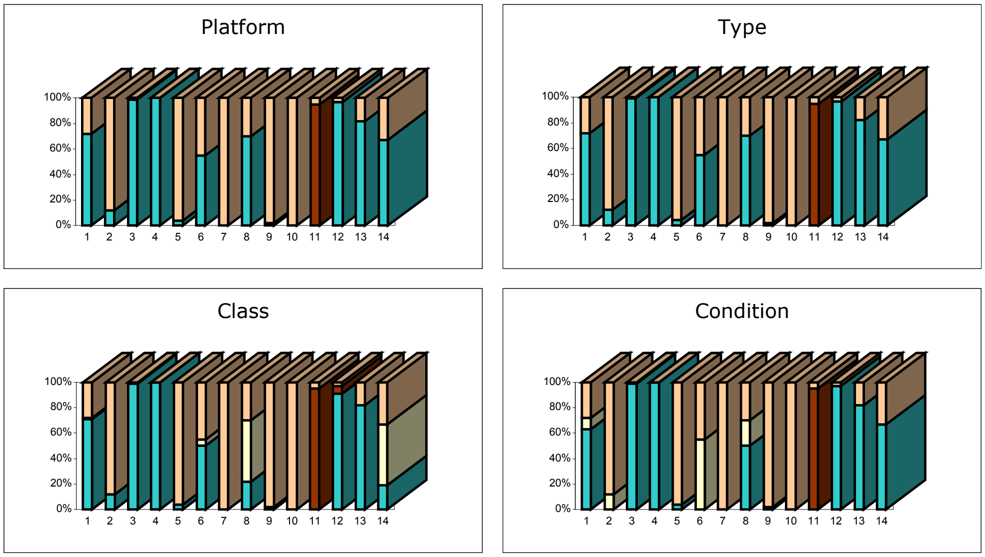


Figure 49: Graphical representation of Table 12.

The results seem terrible when we use this type of SOM for open-set identification method. The orange bars in *Figure49* represent unknown identifications. In *Figure 50*, winning neurons of test vectors are obviously in the middle of a cluster or near of a cluster. We can decide that these vectors belong to that specific cluster visually but in SOM point of view these neurons represent unknown identity.

The majority of unknown identities in overall test forced us to apply GMM in order to homogenize the clusters in topographic map and label the unlabelled neurons. Although it is applicable to convert this collaboration suitable for open-set identification task, we have used this model for closed-set identification tasks.

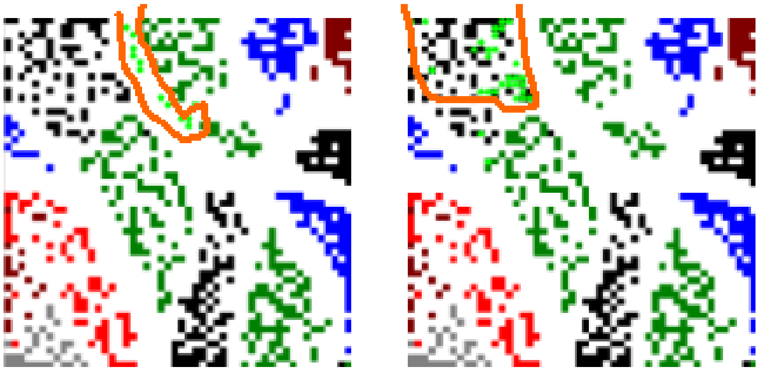


Figure 50: Test results displayed in topographic map. Lime cross marks represent the winning neurons of test pattern.

First we have trained SOM core with feature vectors and had topographic map of SOM, in which there are one x-y coordinates of neuron and one label of it. So we feed GMM with these new generated data vectors, so shifted from high dimensional feature vectors to two-dimensional coordinate vectors for train processes. For test processes, again, we have converted test vectors into coordinate vectors with SOM core and tested these with GMM.

We set the map dimension as “40x40” and made the epoch count 75000. At the end of SOM training process, the teach error was 7,86E-5, which was satisfactory for us. Also maximum GMM cluster count set as 200. *Table 13* shows test results and *Figure 51* gives distinct ideas about the comparisons.

Table 13: Comparison of test results with SOM alone and Combination of SOM and GMM. Tests were made only for “Class” branch of taxonomy.

Test Num	**SOM**		**SOM + GMM**
	Match %	Not Classified	Match %
1	84	15	99
2	45	55	73
3	99	0	99
4	100	0	68
5	5	95	70
6	78	22	83
7	0	100	59
8	20	40	46
9	0	100	100
10	0	100	100
11	0	100	0
12	97	0	100
13	87	0	91
14	20	44	87

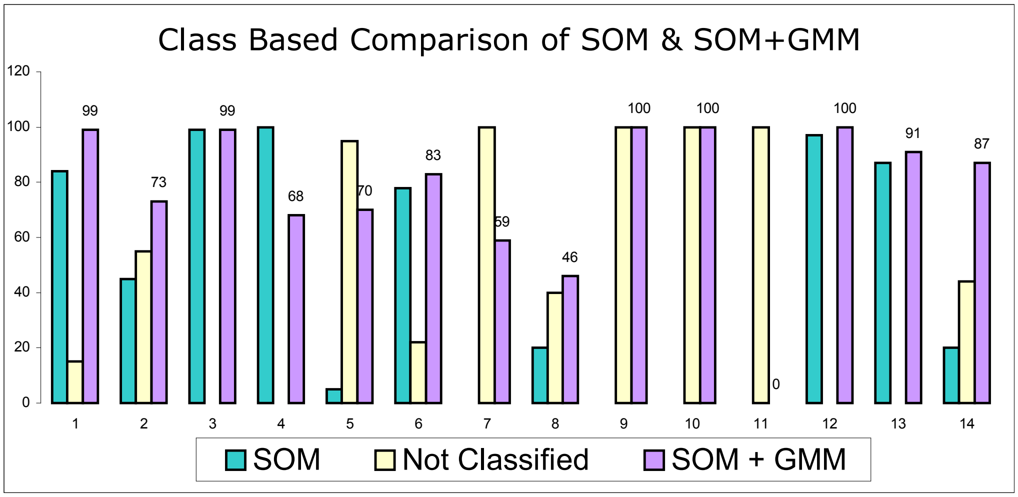


Figure 51: Comparison graph of Table 13.

The results are not as good as BPNN but it is clear that original raw sound data looses information two times more than usual. First in SOM phase and second in GMM' s vector quantization. In spite of this inclination to be erroneous, the results are unexpectedly satisfactory.

5.2.2.3. Gaussian Mixture Model

GMM was used in our experiments as another unsupervised learning algorithm and the results were amazing. After having average successful test results from SOM, GMM were provided unexpectedly successful results from basically clustering algorithm. Similarly above the applications, the model was trained with test vectors and the centroids were labeled with known taxonomical positions of vectors. The model was allowed to assemble up to 200 clusters. But the experiments showed that 130-135 clusters were enough to obtain 95% percent of correct grouping. Also 20-30 iterations were enough to keep clusters stable (suffer in minima).

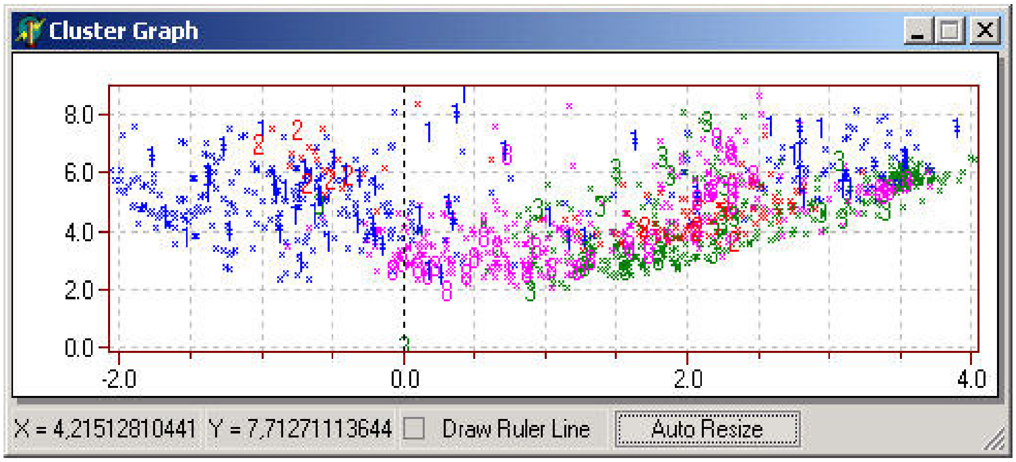
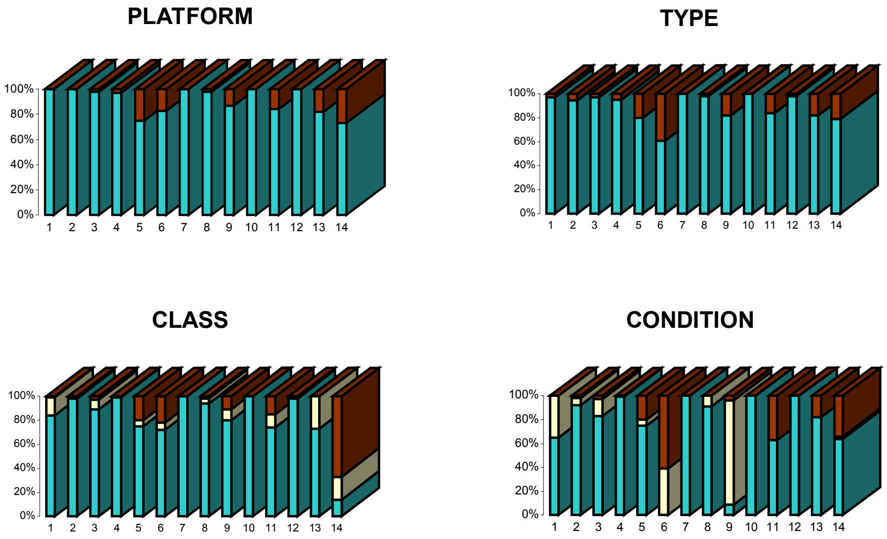


Figure 52: Example of Gaussian Mixture Model clusters.

Table 14: Test results of GMM. Profile I settings were used for Feature Extraction

PROFILE I		PLATFORM			TYPE			CLASS			CONDITION		
Mixture Count		132			140			131			134		
Correct Cover Ratio		96,40%			96,60%			94,70%			93,90%		
Iteration		33			26			31			35		
	Test No	M	CM	WM	M	CM	WM	M	CM	WM	M	CM	WM
	1	97	0	3	100	0	0	84	15	1	65	35	0
	2	100	0	6	100	0	0	98	2	0	92	6	2
	3	97	0	3	98	0	2	89	8	3	83	14	3
	4	95	0	5	97	0	3	99	1	0	99	0	1
	5	80	0	20	75	0	25	75	5	20	75	5	20
	6	61	0	39	83	0	17	72	6	22	0	39	61
	7	100	0	0	100	0	0	100	0	0	100	0	0
	8	98	0	2	98	0	2	94	4	2	91	9	0
	9	82	0	18	87	0	13	80	9	11	9	87	4
	10	100	0	0	100	0	0	100	0	0	100	0	0
	11	84	0	16	84	0	16	74	11	15	63	0	37
	12	98	0	2	100	0	0	98	0	2	100	0	0
	13	82	0	18	82	0	18	73	27	0	82	0	18
	14	79	0	21	73	0	27	14	19	67	64	2	34


PLATFORM

TYPE

CLASS

CONDITION

Figure 53: Graphical representation of Table 14.

Table 15: Test results of GMM. Profile II settings were used for Feature Extraction

PROFILE II		PLATFORM			TYPE			CLASS			CONDITION		
Mixture Count		134			135			131			125		
Correct Cover Ratio		97,80%			97,70%			94,70%			92,60%		
Iteration		25			39			31			38		
	Test No	M	CM	WM	M	CM	WM	M	CM	WM	M	CM	WM
	1	100	0	0	100	0	0	75	25	0	78	22	0
	2	98	0	2	100	0	0	96	4	0	77	13	10
	3	95	0	5	98	0	2	89	9	2	85	10	5
	4	100	0	0	92	0	8	97	0	3	94	0	6
	5	97	0	3	78	0	22	60	37	3	65	22	13
	6	83	0	17	39	0	61	67	0	33	44	0	56
	7	100	0	0	100	0	0	100	0	0	100	0	0
	8	100	0	0	100	0	0	83	13	4	89	11	0
	9	100	0	0	100	0	0	100	0	0	2	96	2
	10	100	0	0	100	0	0	100	0	0	89	11	0
	11	63	0	37	53	0	47	26	37	37	74	0	26
	12	97	0	3	95	0	5	95	2	3	97	2	1
	13	82	0	18	91	0	9	82	0	18	82	0	18
	14	83	0	17	67	0	33	14	84	2	52	0	48

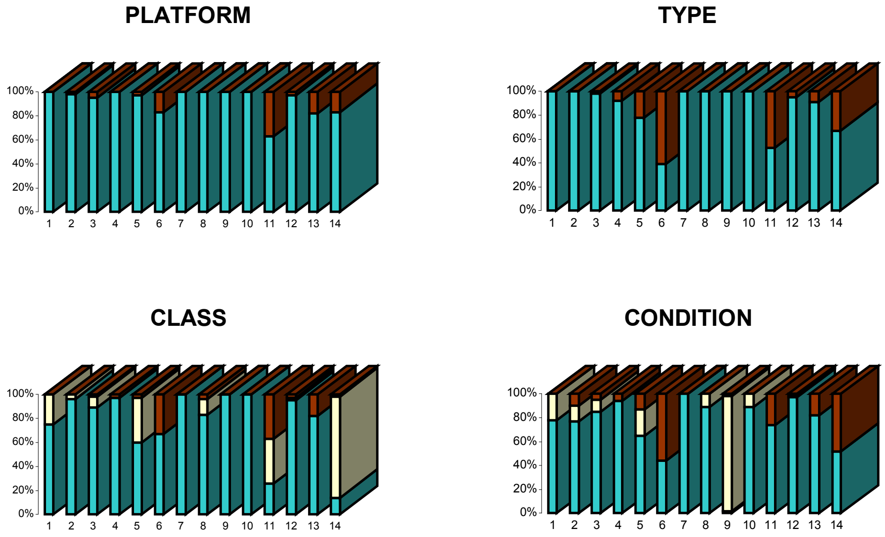


Figure 54: Graphical representation of Table 15.

Table 16: Test results of GMM. Profile III settings were used for Feature Extraction

PROFILE III		PLATFORM			TYPE			CLASS			CONDITION		
Mixture Count		135			134			127			131		
Correct Cover Ratio		98,30%			98,30%			94,00%			95,00%		
Iteration		66			31			41			46		
	Test No	M	CM	WM	M	CM	WM	M	CM	WM	M	CM	WM
	1	100	0	0	100	0	0	97	3	0	69	31	0
	2	100	0	2	100	0	0	92	4	4	98	2	0
	3	99	0	1	100	0	0	98	1	1	89	10	1
	4	98	0	2	97	0	3	97	0	3	95	0	5
	5	75	0	25	65	0	35	78	7	15	72	3	15
	6	100	0	0	100	0	61	100	0	0	0	78	22
	7	100	0	0	100	0	0	100	0	0	100	0	0
	8	100	0	0	100	0	0	87	11	1	89	11	0
	9	84	0	16	96	0	4	76	20	4	13	87	0
	10	100	0	0	100	0	0	100	0	0	100	0	0
	11	47	0	53	68	0	32	42	26	32	68	0	32
	12	98	0	2	98	0	2	91	5	4	100	0	0
	13	91	0	9	91	0	9	82	9	0	100	0	0
	14	35	0	65	54	0	46	8	84	8	77	0	23

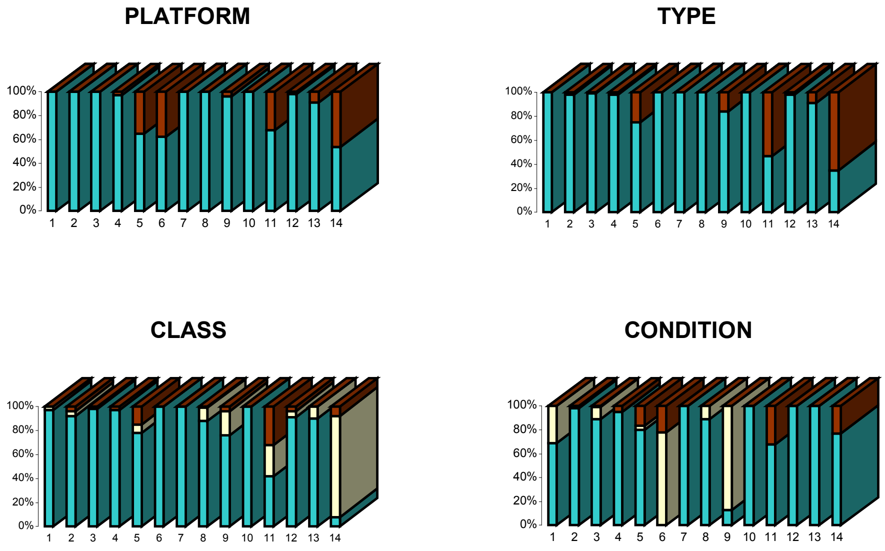


Figure 55: Graphical representation of Table 16.

Correct quantization ratio of GMM depends on the cluster count, if we check out the test tables. This is the property we expected but although we set the maximum mixture count as 200, we didn’t reach that level because of the initial presets of GMM clusters.

As an overall evaluation of GMM with acoustic signatures we have supplied, we can say that

- If we investigate the performance graphs of GMM classifier thoroughly, Profile I feature extraction schema is the best feature extractor as we have determined before.
- Correct cover ratio and mixture counts decrease with increasing variety of identities in relevant taxonomy branches, also prosperity levels does.
- As a disadvantage of GMM, similar clusters can be considered as they all belong to the same identity, so this leads mismatches or close classifications.
- In order to prevent wrong classification, we should construct taxonomy tree with uncorrelated branches. Assume that we have a huge data cloud that result of diesel engines of “Y” and “T” classes. If we test this model for “condition” branch then it is easy to classify, on the other hand, if we test this model for “class” branch, it is very hard to divide this cluster into sub-clusters with GMM. This assumption became definite when we have tested with SOM and GMM together with two-dimensional feature vectors and might be invalid if data lengths of feature vectors are high enough.
- Condition branch seems useless with GMM as we determined before.

CHAPTER 6

CONCLUSIONS AND FUTURE RESEARCH

In this section, conclusions about the research framework were made and the answers of the problem statement were evaluated that we declared in *Chapter 1* as "*How can I develop a computer system that gives the Naval Ships and Submarines the ability of classification and discrimination of surface or subsurface targets using TRN & LOFAR signals.*" Also our experimental results were concluded.

Some clues about possible future extensions, which were not studied in this work, of complete structure of acoustic signature detection were presented also.

6.1. Conclusions

We examined that feature extraction processes definitely affect the complete signature evaluation system. Since underwater acoustic properties are quite different from speech properties studied much more till today, using only usual feature extraction methods may not work properly. Furthermore our experimental results proved this hypothesis.

As we inspected in *Chapter 2*, underwater acoustic information emitted from target platforms has vast variety of signal discrimination properties. A small segment of sound data may give valuable information about target signature sometimes but also whole data should be considered most of the times. The performance evaluation of feature extraction process requires taking into account time variability in Underwater Acoustic Channel (UWAC) that makes the DSP phase of feature extraction process much more difficult due to the adaptive filter requirements. In order to benefit from both properties above, frequency domain interpretation containing time information should be used with large segments of data. At this time wavelet packages come to the stage. This may be called basically data preprocessing.

Application of LPC onto this schema reduces the dimensions of data vectors while keeping the valuable information. Finally, valuable weighted frequency information should be added such as high important low frequency components and less important high frequency components. Mel scaled frequency interpretation helps us to cover this.

We didn’t use PCA for dimension reducing since PCA is used for unsupervised processes. This forces us not to use PCA for BPNN. Also SOM acts as principle component analyzer, there is no need to use it for SOM operations. Also we studied that DCT in MFCC is similar to PCA.

In Profile I feature extraction schema, we tested all above and got better results than the other profiles. Profile II we tested is successful also but not comprises our all requirements in the sense of signature generation. LPC has been basically designed for LTI systems but UWACs are characterized by multipath phenomenon whose characteristics are time varying.

Labeling the feature vectors is as important as the feature vectors themselves. We labeled the vectors and codebooks according to a simple taxonomy tree that was used without consideration of tactical operation requirements of candidate users. For example, we didn’t take the submarine commanders information requirements into account whether he needs the targets platform, type, class or another information. Intelligence reports and sonar operator’s skills on acoustic processing affect those requirements also.

Our experiments showed that platform, type and class branches are suitable but condition branch is not. This is the result of that; condition branch is common property, which lies below the other branches. In order to capture condition properties of target, we should make some modifications on feature extraction phase or completely reconstruct the taxonomy tree. We should make other signal processing operations such as DEMON.

We tested three basic recognition schemas and a combination of them. But the results are quite similar for our test patterns. In the figure below, we see that similarity.

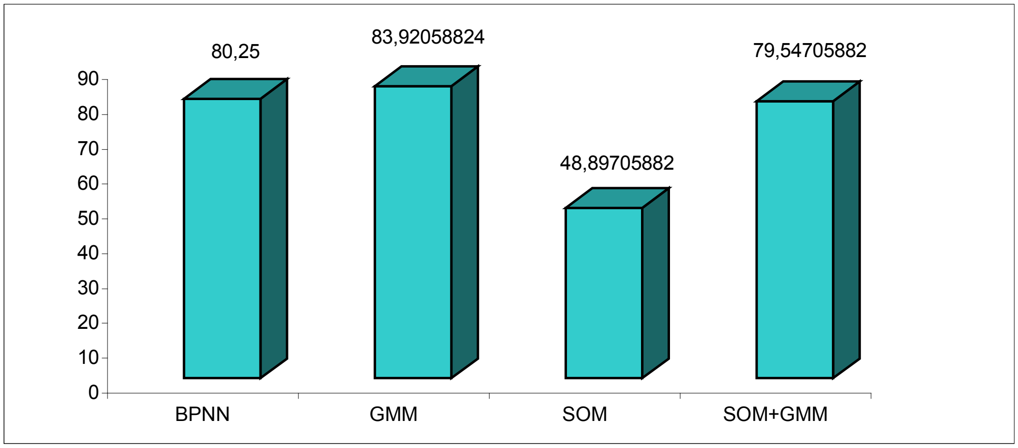


Figure 56: Comparisons of correct classification rates of recognition schemas.

Success rate of system also depends on the decision-making phase but we see that almost all of the schemas we tested are suitable except accustomed use of SOM. Application of a robust decision-making process makes the system powerful.

Using usual SOM method as a part feature extraction instead of recognition, we have better results as indicated in *Figure 56*.

6.2. Future Research

In frequency domain interpretation, it is possible to consider whole data representation as an image containing edges and noises as kind of both salt-pepper and gaussian. Processing this image might let us have better recognition rates.

Application of image processing techniques helps us to recover lost data, filter the useless ripples of frequency spectrum and noise, and sharpen the certain edges representing transient impact noise that is a signature of a certain kind.

Our basic taxonomy tree for whole system is not adequate for practical use of system. A demonstrative taxonomy should be constructed with collaboration of candidate users and some fine-tuning should be made to other phases accordingly.

Since we have tested small amount of data with relevant recognition methods, it was impossible to measure correct computational performance such as database storage capacity, computational load of CPU and RAM, time-consuming rates. As a future work, after having enough data available, performance testing of classifiers should be made and needed ameliorations should be applied.

Although decision-making is beyond the scope of this work, it is indispensable from acoustic signature detection processes. We choose simple decision parameters in our examples such as minimal mean error rate, maximal vector match count or threshold selection.

Recognition methods we used provide considerable inputs to decision-making phase. In order to increase correct matching rates, application of some data mining algorithms might be considered as worthwhile.

REFERENCES

[1] Başaran H., Düzenli Ö., İlgüy C., "*Classification Of TRN And LOFAR Signals From Surface And Subsurface Platforms Using Wavelet-Based Decompositions*", NATO Symposium, SET-079/RSY, La Spezia, Italy, 2004.

[2] Bogert B. P., Tukey J. W., M. J. R. Healy, "*The Quefrency Analysis Of Time Series For Echoes: Cepstrum, Pseudo-Autocovariance, Cross-Cepstrum, And Saphe-Cracking*", Proceedings Of The Symposium On Time Series Analysis.

[3] Brekhovskikh L.M., Lysanov Y.P., "*Fundamentals of Ocean Acoustics*", 3rd ed. Springer NewYork., 2003.

[4] Bullinaria J.A., "*Introduction to Neural Networks*", Birmingham University, 2004, PDF version available at www.cs.bham.ac.uk/~jxb/NN, last access in April 2005.

[5] Coifman R.R., Wickerhauser M.V., "*Entropy_Based Algorithms For Best Basis Selection*", IEEE Trans. Inform. Theory Vol. 38 No. 2 Pp 713 – 719 , 1992

[6] "*Course Notes for Sonar Operators*" and "*Sonar Operator's Handbook*", Submarine Education Center of Turkish Navy.

[7] Cowling M., "*Non-Speech Environmental Sound Classification System For Autonomous Surveillance*", 2004, PDF version avaible at www4.gu.edu.au:8080/adt-root/uploads/approved/adt-QGU20040428.152425/public/ , last access in March 2005.

[8] Damper R., Higgins J., "*Improving Speaker Identification In Noise By Subband Processing And Decision Fusion*", Pattern Recognition Letters 24, 2003.

[9] Daubechies I., "*Ten Lectures on Wavelets*", Society for Industrial and Applied Mathematics, Philadelphia, Pa., 1992.

[10] Deller Jr. J.R., Hansen J.H.L., Proakis J.G., "*Discretetime Processing Of Speech Signals*", Macmillan Publishing Company, 2000.

[11] Ellis D., "*Speech Models*", SAPR, 2001, PDF version available at http://www.ee.columbia.edu/~dpwe/e6820/lectures/, last access in March 2005.

[12] Etter P.C., "*Underwater Acoustic Modeling and Simulation*", 3rd ed., Spon Press, London, 2003.

[13] Fränti P., Kivijärvi J., "*Randomized Local Search Algorithm For Clustering Problem*", University Of Joensuu Department Of Computer Science, Report Series A, 1999 – 5.

[14] Fränti P., Virmajoki O., "*Practical methods for speeding-up the pairwise nearest neighbor method*," Opt. Eng. 40#11#, 2495--2504 #2001# http://citeseer.ist.psu.edu/virmajoki01practical.html

[15] Fränti P., Virmajoki O., and Kaukoranta T. "*Branch-and-Bound Technique for Solving Optimal Clustering*" ICPR, vol. 02, no. 2, p. 20232, 16 2002.

[16] Fränti P., Virmajoki O., Hautomäki V., "*Graph – Based Agglomerative Clustering*", University Of Joensuu Department Of Computer Science, Report Series A, 2003 – 4.

[17] Gaps A., "*An Introduction To Wavelets*", IEEE Computational Science And Engineering, Vol. 2, 1995.

[18] Harris J. "*On The Use Of Windows For Harmonic Analysis With Discrete Fourier Transform*", Proceedings Of The IEEE, Vol. 66, No. 1, 1978.

[19] Huang X., Acero A., Hon H., "S*poken Language Processing - A Guide to Theory, Algorithm, and System Development*", Prentice Hall, 2001.

[20] Ifeachor E., Lewis B., "*Digital Signal Processing – A Practical Approach, 2nd Ed.*", Pearson Education Limited, Edinburgh Gate, 2002.

[21] Kärkkäinen I., Fränti P., "*Stepwise Clustering Algorithm For Unknown Number Of Clusters*", University Of Joensuu Department Of Computer Science, Report Series A, 2002 – 5.

[22] Kärkkäinen I., Fränti P., "*Dynamic Local Search for Clustering with Unknown Number of Clusters*" *icpr*, vol. 02, no. 2, p. 20240, 16 2002.

[23] Kinnunen T., "*Spectral Features For Automatic Text-Independent Speaker Recognition*", Licentiate's Thesis, University Of Joensuu Department Of Computer Science, 2003.

[24] Kinnunen T., Kärkkäinen I., Fränti P., "*Statistical Analysis Of Cepstral Features*", *European Conf. on Speech Communiation and Technology, (EUROSPEECH'2001)*, Aalborg, Denmark, Vol. 4, pp. 2627-2630, September, 2001.

[25] Koç A., "*Acoustic Feature Analysis For Robust Speech Recognition*", MS. Thesis, Boğaziçi University, 2002

[26] Kohonen T., "*Self-Organization and Associative Memory*", Springer Verlag, Berlin, 3rd edition, 1989.

[27] Long, C.J., Datta, S. "*Wavelet based feature extraction for phoneme recognition.*" Proc. of 4th Int. Conf. of Spoken Language Processing, Philadelphia, USA, Vol. 1 (1996) 264-267

[28] Mahmood R. Azimi-Sadjadi, "*Underwater Target Classification In Changing Environments Using An Adaptive Feature Mapping*", IEEE Transactions On Neural Networks, Vol 13, No.5, 2002

[29] Makhaul J., "*Linear Prediction: A Tutorial Review*", Proceedings Of IEEE 64,561-580, 1975.

[30] Malewicz G., Skarbek W., "*Distributed Evolutionary Algorithm For Vector Quantisation In JAVA*", University Of Warsaw, Department Of Mathematics, Computer Science And Mechanics, 2003, PDF version is avaible at cs.ua.edu/~greg/publications/, last access in May 2005.

[31] Matthews J., "*An Introduction To Neural Networks*", Article in http://www.generation5.org/articles.asp, submitted in 31.03.2000

[32] Morgan N., Gold B., "*Cepstrum Analysis Lecture Notes*", University Of California, 1999, PDF version is available at www.icsi.berkeley.edu/eecs225d/spr01/lectures/, last access in March 2005.

[33] Ng, L. C., Gable, T. J., Holzrichter, J. F.Lawrence, "*Speaker Verification Using Combined Acoustic And EM Sensor Signal Processing*", Livermore National Laboratory And University Of California, ICASP 2001 Salt Lake City

[34] Picone J., "*Fundamentals Of Speech Recognition*", Missisipi Satate University, 2002, PDF version available at http://www.isip.msstate.edu/resources/courses/ece_8463, last access in March 2005.

[35] Polikar R., "*The Engineer's Ultimate Guide To Wavelet Analysis*", Ames, IA, 1994, available online at http://engineering.rowan.edu/~polikar/WAVELETS/WTtutorial.html, last access in June 2005

[36] Proakis J.G., Manolakis D.G., "*Digital Signal Processing: Principles, Algorithms, And Applications, 2nd Ed.*", Maxwell Macmillan International, New York, USA, 1992.

[37] Rabiner L.R., Juang B.W., "*Fundamentals Of Speech Recognition*", Prentice-Hall, Upper Saddle River, New Jersey, USA.

[38] Rehagen D., Kirk R., "*The GNNV Project tutorial*", online at http://www.iwu.edu/~shelley/gnnv/index.html, last access in May 2005.

[39] Rocchesso D. "*Introduction to Sound Processing*",2004,236 pages, ISBN 88-901126-1-1

[40] Roch M.,”*Overview Of Pattern Recognition Methods For Speech & Speaker Recognition*”, San Diego State University, PDF version available at www.rohan.sdsu.edu/~mroch/acoustic/slides/, last access in May 2005.

[41] Rose, P., “*Forensic Speaker Identification*” Taylor & Francis, 2002.

[42] Saito N., Coifman RR., “*Local Discriminant Bases*”, Journal Of Mathematical Imaging And Vision Volume 5, Issue 4 (December 1995), Pp 337 – 358, 1995

[43] Sheng Y., “*Wavelet Transform.*”, The Transforms And Applications Handbook., CRC Press, 1996.

[44] Shental N., Bar-Hillel A., Hertz T., And Weinshall D., “*Computing Gaussian Mixture Models With Em Using Equivalence Constraints*”, Leibniz Center For Research In Computer Science Technical Report 2003-43

[45] Tanenbaum A.S., “*Computer Networks 4th Ed.*”, Prentice Hall, Amsterdam, 2003

[46] Tomasi C., “*Estimating Gaussian Mixture Densities With EM – A Tutorial*”, Duke University, PDF version available at http://www.cs.duke.edu/courses/spring05/cps296.1/handouts/EM/, last access in February 2005.

[47] Tzanetakis G., Essl G., Cook P., "Audio Analysis using the Discrete Wavelet Transform " *In. Proc. WSES Int. Conf. Acoustics and Music: Theory and Applications (AMTA 2001)* Skiathos, Greece, 2001

[48] Urick R.J., “*Principles of Underwater Sound*”, 3rd ed., McGraw Hill,USA, 1983.

[49] Valens C., “*A Really Friendly Guide To The Wavelets*”, PDF version available at http://perso.wanadoo.fr/polyvalens/clemens/download/, last access in June 2005.