INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

New Pitch Based Techniques For Speech Enhancement and Speaker Count Determination

by

Michael A. F. Lewis

A dissertation submitted to the Graduate Faculty in Engineering in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York

1998

This manuscript has been read and accepted for the Graduate Faculty in Engineering in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

6/17/98
Date

Chair of Examining Committee

6/22/98
Date

Executive Officer

PROFESSOR MITRA BASU

PROFESSOR MICHAEL CONNER

PROFESSOR SRINIVASA VEMURU

DR. RAVI RAMACHANDRAN

Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

Abstract

New Pitch Based Techniques For Speech Enhancement and Speaker Count Determination

by

Michael A. F. Lewis

Advisor: Professor Joseph Barba

In this thesis, new pitch-based techniques are investigated with the main objective of making speaker recognition applications more robust to the variable noise conditions experienced in the real world. Cochannel interference of speech signals is a common practical problem particularly in tactical communications. We examine the problem of identifying temporal regions or frames as being either one-speaker or two-speaker speech. This identification is important in making automatic speaker and speech recognition systems more robust and is based on feature extraction and subsequent classification as performed in pattern recognition. We propose a new pitch prediction feature (PPF) which is compared with the Linear Predictive Cepstral Coefficients (LPCC) and the Mel Frequency Cepstral Coefficients (MFCC). The results show that the PPF performs better than all the other features for the closed and open set cases. The problem of automatic and accurate determination of the pitch period in noisy environments is also addressed. We propose a new pitch detection algorithm based on an iterative adaptive smoothing approach using a Gaussian Derivative (GD)

filter. An adaptive gaussian derivative pitch detector was developed to work under varying noise conditions, with variable pitch periods and for different speakers. We compare the performance of the Dyadic Wavelet Transform (DyWT) algorithm with our new Adaptive Gaussian Derivative Filter (AGDF) algorithm for pitch detection of synthesized speech under different noise conditions and signal-to-noise ratios. The results show that the AGDF outperforms the DyWT pitch detection scheme at low signal-to-noise ratios for different types of noise. The AGDF and DyWT algorithms are applied to speech enhancement using an Adaptive Comb Filtering (ACF) scheme. The results of the enhanced speech signal are demonstrated and a cepstral distance measure is used to evaluate the speech enhancement algorithms' performance for various signal-to-noise ratios. We show that ACF speech enhancement lowered the cepstral distance of speech in the presence of colored and babble noise. Furthermore, there was a significant decrease in the cepstral distance of speech in the presence of white gaussian noise.

Acknowlegements

I would like to thank Dr. Ravi Ramachandran, my research advisor, for his invaluable contributions to the completion and success of this thesis.

Thanks to Dr. Rolston Jeremiah, Dr. Alvin Garcia, Dr. Lesa Kennedy Lewis and all my colleagues at the City College of New York, Rutgers' CAIP Center, TNetix SpeakEZ and Rowan University.

Thanks to Dr. Ronald Brown, Dr. Joseph Barba, Dr. Richard Mammone and all the other members of the faculty that have contributed to this thesis.

Thanks to the NSF, CASI and Dean Ramona Brown of the PRES program that have supported me through this thesis.

Thanks to my parents and the rest of the family for all their love and support.

Contents

List of Figures

List of Tables

Chapter 1

Thesis Problem and Organization

1.1 Thesis Problem

Speech processing technology has made many great advances in the past two decades. This progress is due to basic advances in speech and language technologies, as well as rapid increases in computer processing power. As a result, there have been many applications developed featuring these technologies. Although these systems operate with a high degree of success, there is still a great deal of work to be done in developing these applications to their full potential. One of the major problems affecting speech processing systems has to do with the lack of robustness of these systems. Robustness in speech processing is often defined as the minimal, graceful degradation in performance due to changes in input conditions caused by microphones,

room acoustics, background or channel noise, different speakers and other small systematic changes in the acoustic signal. Currently, many speech processing systems under conditions of co-channel speaker interference and background or channel noise experience a high level of diminished performance.

The problem of co-channel interference of signals, though a very heavily researched area in the past two decades, is still clearly one of the more formidable problems in signal processing. In the field of communication in particular, co-channel speech is obviously a major concern. How can the voice of a given speaker be separated from the voice of other interfering speakers without losing the intelligibility of the primary speaker? How can we even recognize what can be classified as co-channel speech? Not surprisingly, the results to present have not been very encouraging.

The interference in the signals may be introduced in the communication channel, at some point during transmitting and receiving end. In tactical communication systems where there are multiple signals transmitted over a single channel this problem is common. It is also possible that interference may be introduced at the transmission site itself. This is often the case if the microphone at the transmitting end is not acoustically isolated, in which case all background noises, including voices would be transmitted along with the primary speaker. This scenario is often exemplified in speaker phones and other hands-free communication devices. No matter where the interference is introduced, the end result is a slightly distorted signal that may be

described as the composite signal consisting of the voices of multiple people speaking simultaneously.

The human brain's ability to effortlessly recover desired speech is known as the "cock-tail party effect". The brain requires binaural data for this effect to be performed. Speech received in a noisy environment over a communication channel is monaural, and can be difficult if not impossible to separate [25]. This distortion of the speech signal by other interfering speakers also affects the ability of automatic speech processing systems, such as speech and speaker recognition systems, to perform effectively and efficiently. Obviously, it would be desirable to suppress the interference due to the interfering speakers, so that the speech of the target speaker can be more easily understood, either by a human listener or a machine. This task is commonly referred to as *co-channel interference reduction*. However, at other times it is desirable to recover not only the speech of the target speaker, but also that of the interfering speakers. In this case, the task is commonly referred to as *co-channel speaker separation*.

In order to achieve the ultimate goal of developing a robust and accurate automatic co-channel speaker separation (CCSS) system for separating a corrupted speech signal into its constituent signals, it is necessary to first obtain as much information as possible about that signal. For instance, the ability to recognize co-channel speech automatically (i.e. *speaker count determination*) and to determine the level of mixing

or corruption of the signal (i.e. *speaker mixing ratio* or *voice to voice ratio*) would significantly enhance the performance of these CCSS systems. In other applications such as speaker identification and/or verification it may not be necessary to separate interfering speakers. It may be sufficient to perform the necessary tasks on frames that have been previously identified as 'clean' of interfering speakers. In this work, we will not attempt to solve the entire co-channel speaker separation or reduction problem. Instead, we have taken on the smaller but yet formidable task of speaker count determination for application in speaker ID systems.

The other major problem is speech degraded by background or channel noise. It can be easily argued that variability of the physical environments and the speech signal, are the most immediate problems facing speech processing engineers today. How can the speech signals be restored once corrupted by noise from any source? How will a speaker recognition system which does well in a laboratory environment perform with cellular phones? Any method that can enhance the speech signal while preserving the intelligibility of the speech signal or just increasing the signal-to-noise ratio will be an important development in speech processing.

An important approach to the problem of speech degraded by background or channel noise lies with the quasi-periodic nature of the speech waveform which corresponds to narrow harmonically spaced bands of energy in the frequency domain. In this work, we have developed a novel approach for determining the periodicity of a speech signal

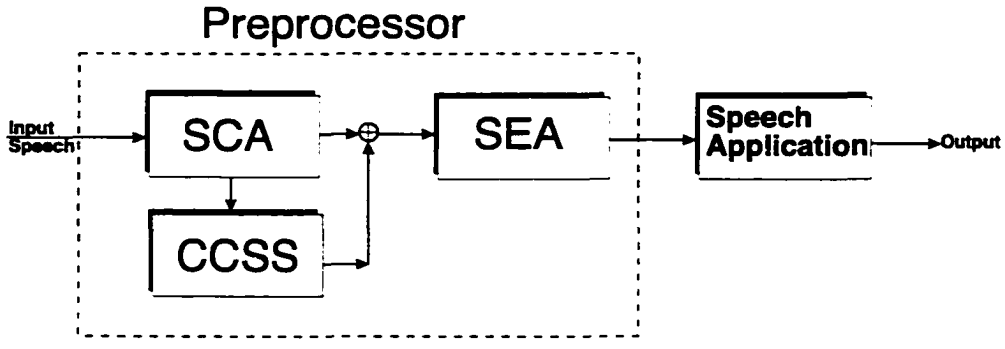


Figure 1.1: The use of Preprocessors with speech Applications

in noisy environments based on an adaptive smoothing scheme. From the accurate determination of the pitch information in a signal in the presence of noise, it is possible to perform enhancement by using an adaptive comb filtering approach.

Our ultimate goal is for the Speaker Count Algorithm (SCA) to be used in conjunction with a Speech Enhancement Algorithm (SEA) for applications in speaker recognition, speech recognition, language identification and other speech processing applications as illustrated in (See Fig. 1.1).

1.2 Thesis Organization

In this thesis, novel robust pitch based techniques to enhance speech primarily for applications in speaker ID are presented. The research can be organized into three main section:

1. In Chapter 3, the speaker count problem is addressed. The problem of identifying temporal regions or frames as being either one-speaker or two-speaker speech. This identification is important in making automatic speaker and speech recognition systems more robust and is based on feature extraction and subsequent classification as is done in pattern recognition. The research looks into both the closed set problem where the identity of the two interfering speakers are known *apriori* and the more difficult open set problem where the identities are not known (speaker independent). For the feature extraction step, we use a new pitch prediction feature (PPF) which is compared with the Linear Predictive Cepstral Coefficients (LPCC) and the Mel Frequency Cepstral Coefficients (MFCC). The features are computed and classified on a frame by frame basis. We compare the performance of two classifiers, namely, the neural tree network (NTN) and vector quantizer (VQ).

2. In Chapter 4, a novel robust pitch detection method is investigated. The new pitch detection algorithm based on an iterative adaptive smoothing approach uses a gaussian derivative (GD) filter which is the sum of a zeroth and second order hermite function. An adaptive gaussian derivative pitch detector was developed to work under varying noise conditions, with variable pitch periods and for different speakers. We compare the performance of the Dyadic Wavelet Transform (DyWT) algorithm with our new Adaptive Gaussian Derivative Filter (AGDF) algorithm for pitch detection of synthesized speech under different

noise conditions and signal-to-noise ratios. In addition, the AGDF algorithm is demonstrated on real and conversational speech.

3. In Chapter 5, the pitch determination methods discussed in Chapter 4, are applied in an adaptive comb filtering method to enhance the speech. Both the DyWT and AGDF algorithms are used for the first time for speech enhancement. Enhanced speech segments are demonstrated to show the effectiveness of both algorithms. The performance of the algorithms are evaluated using the cepstral distance measure for various conditions of noise and signal to noise ratios.

In Chapter 2, the theoretical background information used in Chapters 3, 4 and 5 are presented. In particular, Chapter 2 gives the information on the features and pattern recognition techniques utilized in the thesis. These tools are the basis for all the algorithms developed in this work. In Chapter 6, all the thesis accomplishments are summarized. Several proposals for the application of the preprocessing techniques developed to Speaker ID systems are also outlined for future work. In addition, there are also three appendices on additional information referenced throughout the thesis.

Chapter 2

Background on Speech Production and Processing Techniques

2.1 Introduction

1 Speech Production

The human vocal system is comprised of the vocal tract, the nasal tract and lungs [41]. The vocal tract begins at the opening of the vocal cords or glottis, and ends at the lips. The vocal tract consist of the pharynx which is the connection from the esophagus to the mouth or oral cavity. The nasal tract begins at the velum, a trap-

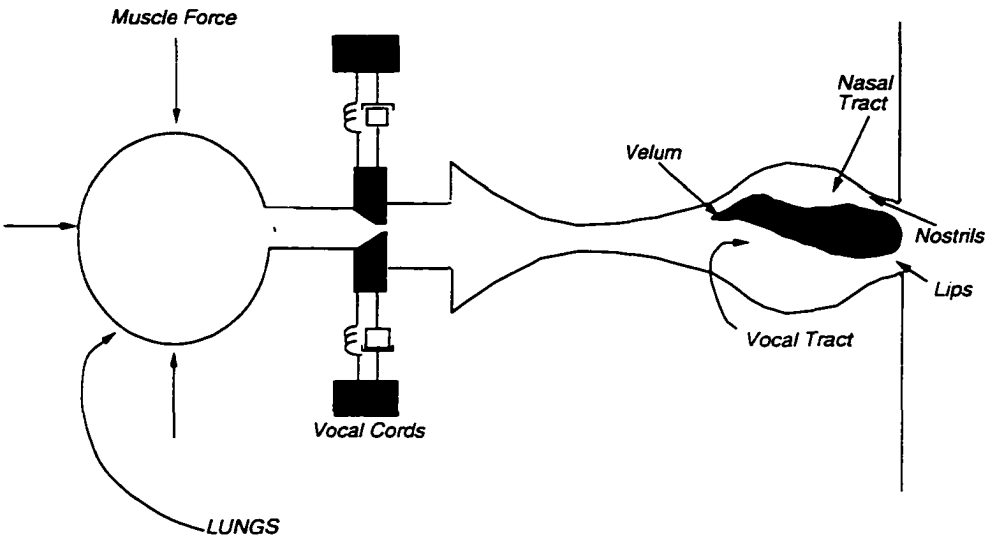


Figure 2.1: A schematic of the vocal tract

door like mechanism at the back of the mouth cavity and ends at the nostrils. When the nasal tract is lowered, it is acoustically coupled to the vocal tract to produce the nasal sounds of speech. A schematic of the vocal tract is shown in Fig. 2.1 [2].

In the production of speech, which can be characterized as acoustic waves, the lungs forces air through the vocal and nasal tracts. The tensed vocal cords within the larynx are caused to vibrate by the air flow. The air flow is chopped into quasi-periodic pulses which are then modulated in frequency when passed through the pharynx, throat, the mouth and nasal cavity [4]. The various sounds that are produced occur due to different articulations such as the position of the jaws, lips, velum, etc.

Speech sounds can be classified into three distinct classes according to their mode of

excitation [41]. Voiced sounds are produced when air is forced through the glottis over a tensed vocal cord which vibrates in a relaxed oscillation mode. The end result is quasi-periodic pulses which excite the vocal tract (e.g. oven). Unvoiced sounds results when the vocal cords are relaxed. In order to produce sound in this case, the air flow must pass through a constriction in the vocal tract, thereby becoming turbulent and produces broad-spectrum noise-like waveforms known as fricatives or unvoiced sounds (e.g. shame) . Another type of sound is known as plosive or mixed sounds and results from making a complete closure, usually towards the front of the vocal tract, building up pressure behind the closure, and abruptly releasing it. The wave appears to be small amplitude followed by a burst of noise-like waveform(eg. victory).

Language can be described by a set of distinct speech sounds known as phonemes. American English has a list of 42 phonemes [4]. These phonemes are classified into four main groups: vowels, diphthongs, semi-vowels, and consonants. These groups are further divided into subgroups based on the area function of vocal tract or place of articulation within the vocal tract.

A more general form of classification of phonemes have been to put them into classes of continuant and non-continuant sounds. Continuant sounds are produced by a fixed non-varying vocal tract configuration which is excited by an appropriate source. The class of Continuant sounds consist of vowels, the fricatives, and the nasals.

The remaining sounds: diphthongs, semi-vowels and affricatives are produced by a changing vocal tract configuration and can be classified as non-continuant sounds. From the description, it is clear that speech can be classified as a variable and non-stationary signal.

2 Models for Speech Production

In order to understand the features of the speech signal, it is necessary to study the models for production of speech. Generally, it is known that a dynamically varying vocal tract imposes its resonance upon the excitation so as to produce different sounds of speech. Therefore, in order to produce a speech-like sound it is necessary that there be a mode of excitation and that the resonance properties of the system change with time, remaining fixed for periods of 10-20ms.

The resonances of the vocal tract can be directly related to the formant information in the speech signal. The transfer function of the vocal tract is denoted by $V(z)$, which represents the z-transform of the discrete-time filter which models the frequency response of the vocal/nasal tract system. The all pole model is a very good representation of vocal tract effects for a majority of sounds, however, nasals and fricatives require both resonances and anti-resonances, i.e. poles and zeros. Coefficients of the denominator of V(z) are real, therefore the roots of the polynomial will either be real or occur in complex conjugate pairs.

A typical complex resonant frequency of the vocal tract is

$$s_k s_k^* = -\sigma_k + / - j2 * F_k. \tag{2.1}$$

The bandwidth of the vocal tract resonance is $2\sigma_k$ and the center frequency is $\Omega_k = 2\pi * F_k$. The complex natural frequencies of the human vocal tract are all in the left half of the s-plane which ensures that the system is stable. This means that all the poles are within the unit circle.

Typically, V(z) is modeled as a strictly autoregressive (AR) filter whose parameters are determined by some linear predictive coding (LPC) analysis of the speech signal. In general, however, the vocal tract can be represented by an autoregressive moving average (ARMA) model, similar to Fujimura in [6], consisting of m poles and n zeros of the following form to represent the vocal tract model:

$$V(\omega) = \prod_{k=1}^{m} \frac{1 - 2e^{-\alpha_k T}\cos(\omega_k T) + e^{2\alpha_k T}}{1 - 2e^{-2\alpha_k T}\cos(\omega_k T)e^{-j\omega} + e^{-2\omega_k T}e^{-2j\omega}} \times \prod_{l=1}^{n} \frac{1 - 2e^{-\beta_l T}\cos(\xi_l T)e^{-j\omega} + e^{2\beta_l T}e^{-j2\omega}}{1 - 2e^{-2\beta_l T}\cos(\xi_l T) + e^{-2\beta_l T}} \tag{2.2}$$

where T is the sampling rate, and α_k and β_l are the bandwidths corresponding to the center frequencies of the poles and zeros, ω_k and ξ_l, respectively, of the vocal tract.

Another effect that is taken into consideration is the radiation effects or pressure of the lips. The pressure can be related to a high-pass filter. A reasonable approximation is $R(z) = R_0(1 - z^{-1})$ where R_0 is a constant.

In order to produce sound it is first necessary to have a source. This basically means supplying a train of unit impulses which are spaced by the desired fundamental frequency for voiced signals, or in the case of unvoiced signals, random noise generation. For quasi-periodic speech signals, the excitation is caused by a linear system whose impulse response has the desired glottal wave shape. A gain parameter controls the intensity of the voiced excitation. In [16], it was found that glottal pulse waveform can be represented by the waveform shown inFig. 2.2:

$$g_l(t) = \begin{cases} \frac{1}{2}[1 - \cos(\frac{\pi t}{t_1})] & \text{if } 0 \leq t \leq t_1 \\ \cos(\frac{\pi(t-t_1)}{2t_2}) & \text{if } t_1 \leq t \leq t_1 + t_2 \\ 0 & \text{elsewhere} \end{cases}$$

The variables t_1 and t_2 determines the width of the glottal excitation.

The effect of the glottal pulse in the frequency domain is equivalent to a low pass filtering effect which means that the z-transform has only zeros. For unvoiced sounds all that is required is the source of random noise and a gain parameter to control the intensity.

In the case of voiced signals, all the transfer function of the speech production system-

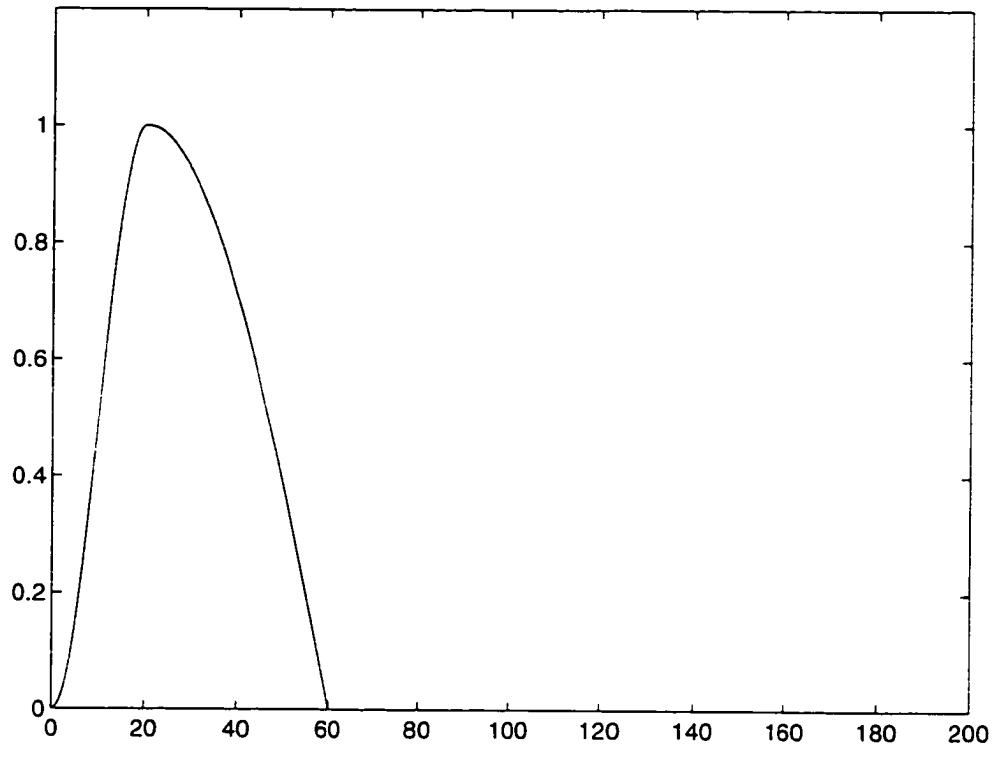


Figure 2.2: The Glottal Pulse

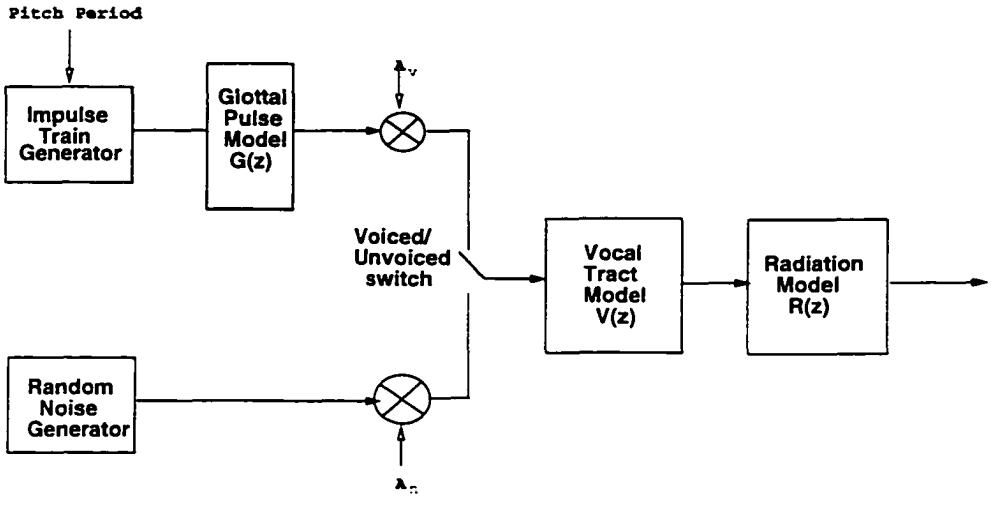


Figure 2.3: A schematic of the linear vocal tract model for speech

radiation model, glottal excitation model and vocal tract model- are lumped together into a single transfer function model:

$$H(z)=G(z)V(z)R(z)$$

The complete model for speech is shown in Fig. 2.3. As mentioned previously, the excitation source for the voiced speech is quasi-periodic, therefore an impulse train generator is used to produce glottal pulses which are separated by a fundamental frequency of f_0 or the pitch[1]. To generate unvoiced signals, on the other hand, a random noise generator is used. The resulting magnitude spectrum is roughly "white" or flat. The parameters A_v and A_n represent the gain factors for the voiced and

[1]Generally, the pitch is used to refer the perception of the fundamental frequency f_0, however, it is used here interchangeable with fundamental frequency

unvoiced excitation sources, respectively, and the switch toggles between the two sources, depending on whether or not the modeled speech is voiced or unvoiced. When the excitation signal propagates through the vocal tract system, the resulting signal has a spectrum whose gross spectral shape is dictated by the frequency selectivity of the system, and whose fine spectral detail is governed by the type of excitation signal.

2.2 Linear Prediction Analysis

Linear Predictive Analysis is the basis for many speech processing applications. This signal processing front-end allows for some form of parameterization of the speech waveform. Ultimately, it means that there is a considerable less information rate to process which are key to many applications. From this parameterization of the signal, it is possible to derive many features. Many of these features are more robust to noise than the LPC coefficients and are utilized in many applications including speaker recognition systems.

In the previous section, a general ARMA model of the speech signal was presented. Typically, however, speech is modeled by an autoregressive or linear prediction transfer function [4] . From this linear prediction model, it is possible to derive a difference

equation for synthesizing the speech samples $s(n)$ which can be expressed as

$$s(n) = \sum_{k=1}^{p} a_k s(n-i) + Gu(n). \tag{2.3}$$

The interpretation is that given p samples at time n, $s(n)$ can be approximated as a linear combination of the past p samples as expressed by relationship in equation (2.3). The coefficients a_k are assumed constant over the analysis frame and the gain factor is ignored to allow the parameterization to be independent of signal intensity. The basic problem of linear prediction analysis is to determine the coefficients a_k of the speech signal, so that the spectral properties of the digital filter in equation 2.3 matches the speech waveform within the analysis window. Since the speech signal varies over time, it is necessary to apply short time analysis techniques to obtain the coefficients. The *pth* order polynomial linear predictor polynomial is

$$P(z) = \sum_{k=1}^{p} a_k z^{-k} \tag{2.4}$$

The prediction error, $e(n)$, is defined as

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k). \tag{2.5}$$

where $\tilde{s}(n) = \sum_{k=1}^{p} a_k s(n-k)$ is the estimate of the speech signal. The error transfer function can then be defined as

$$A(z) = \frac{E(z)}{S(z)} = 1 - \sum_{k=1}^{p} a_k s(n-k). \tag{2.6}$$

In a direct comparison, with equation (2.3), it is clear that the transfer function of the system is equivalent to the inverse of the prediction error filter, i.e.

$$H(z) = \frac{G}{A(z)}. \tag{2.7}$$

In order to determine the best coefficients a_k of the system function, the general approach has been to obtain the predictor coefficients that will minimize the mean-squared prediction error over a short segment of the speech waveform. The short-term speech and error segments at time n is defined as

$$\begin{aligned} s_n(n) &= s(n+m), \\ e_n(n) &= e(n+m) \end{aligned} \tag{2.8}$$

where The short-time average prediction error is defined as

$$E_n = \sum_m e_n^2(m) = \sum_m [s_n(m) - \sum_{k=1}^{p} a_k s(m-k)]^2 \tag{2.9}$$

In order to find the prediction coefficients, the error E_n is differentiated with respect to a_k and set to zero. This gives the relationship

$$\sum_m s_n(m-i)s_n(m) = \sum_{k=1}^{p} \hat{a}_k \sum_m s_n(m-i)s_n(m-k). \tag{2.10}$$

where $\hat{a}_k$ are the values of a_k that minimizes the error E_n. In addition, the term

$\sum_m s_n(m-i)s_n(m-k)$ can be viewed as the short-term covariance of $s_n(m)$, i.e.,

$$\phi_n(i,k) = \sum_m s_n(m-i)s_n(m-k) \tag{2.11}$$

We can express the equation (2.10) by

$$\sum_{k=1}^{p} a_k \phi_n(i,k) = \phi_n(i,0) \qquad \text{for i=1,2, ...p} \tag{2.12}$$

From the above relationship, the set of p equations can be used to determine the predictor coefficients that would minimize the mean squared error. Substituting the above relationship into the mean squared error relationship in equation (2.9), we can express E_n as

$$E_n = \phi_n(0,0) - \sum_{k=1}^{p} \hat{a}_k \phi_n(0,k) \tag{2.13}$$

Therefore, the total minimum error consists of the fixed component, and a component which depends on the predictor coefficients. In order to solve for the optimum predictor coefficients, $\hat{a}_k$, $\phi_n(i,j)$ must be computed for $1 \leq i \leq p$ and $0 \leq k \leq p$. The resulting p equations must be solved simultaneously. The computation however, is strongly dependent on m used in defining the both the section of speech for analysis and the region over which the mean squared error is computed. The method most often used to define the range for speech is the autocorrelation method [41]. In this

method, the limits on m which defines the speech segment, $s_n(m)$, is

$$s_n(m) = \begin{cases} s(n+m) \cdot w(m) & 0 \leq m \leq N-1 \\ 0, & \text{otherwise} \end{cases}$$

where $w(m)$ is a finite window of length N that is identically zero outside the interval $0 \leq m \leq N-1$ and $s(n+m)$ is a segment of the speech signal. Based on these facts, the limits of summation of the prediction error for a p^{th} order predictor will be non-zero over the interval $0 \leq m \leq N-1+p$. Thus, the autocorrelation case has an error expressed as

$$E_n = \sum_{m=0}^{N+p-1} e_n^2(m) \tag{2.14}$$

The covariance can then be expressed as

$$\phi_n(i,j) = \sum_{m=0}^{N-p+1} s_n(m-i)s_n(m-k) \begin{cases} 1 \leq i \leq p \\ 0 \leq k \leq p \end{cases} \tag{2.15}$$

Since equation (2.15) is only a function of $i-k$, the covariance function reduces $\phi_n(i,j)$ to the simple autocorrelation function, i.e.

$$\phi_n(i,j) = r_n(i-k) = \sum_{m=0}^{N-1-(i-k)} s_n(m)s_n(m+i-k) \tag{2.16}$$

Since the autocorrelation function is symmetric, i.e. $r_n(-k) = r_n(k)$, the LPC

equations can be expressed as

$$\sum_{k=1}^{p} \hat{a_k} R_n(|i-k|) = R_n(i) \qquad \text{if } 1 \leq i \leq p \tag{2.17}$$

These equations can also be expressed as a $p \times p$ matrix of autocorrelation values that are Toeplitz in form:

$$\begin{bmatrix} r_n(0) & r_n(1) & r_n(2) & \cdots & r_n(p-1) \\ r_n(1) & r_n(0) & r_n(1) & \cdots & r_n(p-2) \\ r_n(2) & r_n(1) & r_n(0) & \cdots & r_n(p-3) \\ \vdots & & & \vdots & \\ r_n(p-1) & r_n(p-2) & r_n(p-3) & \cdots & r_n(0) \end{bmatrix} \begin{bmatrix} \hat{a_1} \\ \hat{a_2} \\ \hat{a_3} \\ \vdots \\ \hat{a_p} \end{bmatrix} = \begin{bmatrix} \hat{r_n}(1) \\ \hat{r_n}(2) \\ \hat{r_n}(3) \\ \vdots \\ \hat{r_n}(p) \end{bmatrix}$$

These equations can be solved by several methods. The most computationally efficient algorithm for solving these equation is the Levinson-Durbin algorithm which is outlined in the next section.

From the solution of the coefficients $\hat{a_k}$, it is now possible to solve for H(z), the vocal tract information. This is considered a naturally robust feature for speech since it may be invariant to noise and other channel effects on the speech signal.

2.3 LPC Derived Feature Vectors

From the predictor coefficients it is now possible to develop several feature vectors. These transformation of the LPC coefficients of the speech signals are utilized in later chapters in this thesis.

1 PARCOR coefficients

The PARCOR or reflection coefficients can be calculated for a section of speech using the Levinson-Durbin algorithm [4], as outlined below:

$$E^0 = r(0) \tag{2.18}$$

$$k_i = \frac{r(i) - \sum_{j=1}^{L-1} \alpha_j^{(i-1)} r(|i-j|)}{E^{(i-1)}} \qquad 1 \leq i \leq p \tag{2.19}$$

$$\alpha_i^i = k_i \tag{2.20}$$

$$\alpha_j^i = \alpha_j^{(i)} - k_i \alpha_{(i-j)}^{(i-1)}. \tag{2.21}$$

$$E^{(i)} = (1 - k_i^2) E^{(i-1)} \tag{2.22}$$

The set of equations are solved recursively for $i = 1, 2, \cdots, p$ and the solution is given by the LPC coefficients $a_m = \alpha_m^p$ for $1 \leq m \leq p$. and k_m are defined as the reflection coefficients .

2 Area Ratios

The reflection coefficients can be also be related to the cross-sectional areas of a non-uniform acoustic tube which models the vocal tract of the speech production system. The vocal tract can be approximated by stacking together p equal length cylindrical sections, each with a constant cross-sectional area $A_m, m = 1, 2, \cdots, p$. When air travels through these sections of unequal areas, wave reflection will occur at the boundaries of these tubes giving rise to the reflection coefficients. These coefficients were derived by the Levinson-Durbin procedure described in the previous section.

The area ratio is defined as

$$g_m = \frac{(1 - k_m)}{(1 + k_m)} = \frac{A_{m+1}}{A_m} \qquad m = 1, 2, \cdots, p \tag{2.23}$$

Note, by taking the logarithm of both sides, a new parameter known as the *Log Area Ratio* (LAR) can be derived.

3 Line Spectral Frequencies

Another parameterization of the LP coefficients is the Line Spectral Frequencies or LSFs. The LSFs are basically the roots of two polynomials based on the inverse filter

A(z) [28]. They can be expressed as

$$\begin{aligned} P(z) &= A(z) + z^{-(p+1)}A(z^{-1}) \\ Q(z) &= A(z) - z^{-(p+1)}A(z^{-1}) \end{aligned} \tag{2.24}$$

These two polynomials are equivalent to artificially augmenting the *pth* section non-uniform acoustic tube (see section on PARCOR coefficients) with an extra section that is either completely closed ($area = 0$) or completely open ($area = \infty$) [4].

The properties of the polynomials are such that $A(z)$ is minimum phase if, and only if,

1. The zeros of $P(z)$ and $Q(z)$ are on the unit circle.
2. The zeros are simple (no multiples).
3. The zeros of interlace.

Generally, the LSF have properties similar to the formant frequencies and bandwidths. The two main properties are that the LSF have localized spectral sensitivity and two closely spaced LSF may indicate the possible presence of a formant.

4 Formants

The formant feature almost has a direct relationship to the LPcoefficients. The formants can be derived by two ways from the LP coefficients. Firstly, by factoring the $p/2$ poles of the predictor polynomial described in Eqn. 2.4. From this information the spectral shaping poles can be matched to the formants by eliminating the poles with extremely large bandwidths. Secondly, the formants can be estimated by taking the LPC derived spectra and applying a peak picking method.

The inherent disadvantage to using LPC method is that the all-pole model is used to model the speech. Nasals sounds or nasalized vowels contain both zeros and poles. When the formant information related to nasals is derived from LP analysis based on the all-pole model, it is unclear how approximation affects the correctness of the result. Another observation is that frame size and frame position have been shown to affect the formant information. Therefore, information related to the bandwidth may also be affected.

5 Cepstrum

Some other important parameter that are derived from the LPC parameters are the cepstral coefficients and their variants. If a signal $x(n)$ which has a z-transform $X(z)$

is considered. then the most general form of representation of that signal is

$$X(z) = G\frac{\prod_{k=1}^{M_i}(1 - a_k z^{-1}) \prod_{k=1}^{M_o}(1 - b_k z)}{\prod_{k=1}^{N_i}(1 - c_k z^{-1}) \prod_{k=1}^{N_o}(1 - d_k z)} \tag{2.25}$$

Where $|a_k|, |b_k|, |c_k|$ and $|d_k|$ are all less than unity. From the above equation, it can be observed that the terms containing a_k and c_k are poles and zeros which lie within the unit circle, whereas the terms containing b_k and d_k all lie outside the unit circle.

The cepstrum is defined as the inverse z-transform of the logarithm of the power spectrum of the signal $x(n)$. Therefore, by taking the power series expansion of the terms it can be shown that the complex spectrum from equation (2.25) can be expressed as

$$c_n = \begin{cases} \log G & n = 0 \\ \sum_{k=1}^{N_i} \frac{c_k^n}{n} - \sum_{k=1}^{M_i} \frac{a_k^n}{n} & n > 0 \\ \sum_{k=1}^{M_0} \frac{b_k^{-n}}{n} - \sum_{k=1}^{N_0} \frac{d_k^{-n}}{n} & n < 0 \end{cases} \tag{2.26}$$

The properties of the above equations will have the following properties [3]

1. The sample $c(0)$ is the natural logarithm of the gain.

2. The poles and zeros of $X(z)$ inside the unit circle contribute only to the causal part of $c(n)$ starting at $n = 1$.

3. The poles and zeros of $X(z)$ outside the unit circle contribute only to the anticausal part of $c(n)$.

4. The cepstrum is causal if, and only if $X(z)$, is minimum phase.

5. The cepstrum is anticausal if, and only if $X(z)$, is minimum phase.

6. The cepstrum $c(n)$ decays as fast as $1/|n|$ as n approaches ∞ and $-\infty$.

7. The cepstrum has infinite duration whether $x(n)$ is of finite or infinite duration.

8. If $x(n)$ is real, $c(n)$ is real.

In the particular case of the *minimum phase* all-pole LP model for speech discussed in equation 2.3, where $z = z_i$ are the poles of the model inside the unit circle, the cepstrum equations in equation 2.26 reduces to

$$c_n = \begin{cases} \log G & n = 0 \\ \frac{1}{n}\sum_{k=1}^{p} z_i^n & n > 0 \\ 0 & n < 0 \end{cases} \tag{2.27}$$

It is then possible to develop a recursive relationship between the predictor coefficients

$$\begin{aligned} c_0 &= \ln \sigma^2 \\ c_m &= a_m + \sum_{k=1}^{m-1} (\frac{k}{m}) c_k a_{(m-k)}, \qquad 1 \leq m \leq p, \\ c_m &= \sum_{k=1}^{m-1} (\frac{k}{m}) c_k a_{(m-k)}, \qquad m > p \end{aligned} \tag{2.28}$$

where σ^2 is the gain term in the LPC model.

The cepstral coefficients have been proven to be much more robust and reliable than LPC, PARCOR and other features, especially in features related to the speaker and speech recognition. In addition, there are several other modified cepstral coefficients, e.g. cepstral derivative coefficients and cepstral mean subtraction coefficients, that can be derived from the cepstral coefficients.

6 Cepstral Weighting

The basic idea behind cepstral weighting is to account for the sensitivity of the low-order cepstral coefficients to overall spectral slope and the sensitivity of the high-order cepstral coefficients to noise [65]. Weighting is accomplished by multiplying $c_{lp}(n)$ by a window $w(n)$ and using the weighted cepstrum as the feature vector. This weighting operation is also known as liftering. The first consequence of liftering is in extracting a finite dimensional feature vector from an infinite duration $c_{lp}(n)$. Also, careful choices of $w(n)$ enhance robustness. There are several schemes of weighting which differ in the type of cepstral window $w(n)$ that is used. The simplest one is the rectangular window as given by

$$w(n) = \begin{cases} 1 & n = 1, 2, ..., L \\ 0 & \text{otherwise} \end{cases} , \tag{2.29}$$

where L is the size of the window. The first L samples, which are the most significant due to the decaying property, are kept. Other forms of $w(n)$ include *quefrency*

liftering (or linear weighting) where

$$w(n) = \begin{cases} n & n = 1, 2, \ldots, L \\ 0 & \text{otherwise} \end{cases}, \tag{2.30}$$

and *bandpass liftering* (BPL) [65][4]] where

$$w(n) = \begin{cases} 1 + \frac{L}{2}\sin(\frac{n\pi}{L}) & n = 1, 2, \ldots, L \\ 0 & \text{otherwise} \end{cases}. \tag{2.31}$$

The frequency liftering weights each individual cepstral component by its index n thereby downplaying the lower order components. The BPL weights a cepstral sequence by a raised sinusoidal function so that the lower and higher order components are deemphasized. Note that the weighting schemes described are fixed in the sense that the weights are only a function of the cepstral index and have no explicit bearing on the instantaneous variations in the cepstrum that are introduced by different environmental conditions (like noise and channel effects).

2.4 Mel Frequency Cepstral Coefficients (MFCC)

The perception of sound by humans of either pure tones or for speech signals have been shown to follow a nonlinear scale. This has led to the definition of what is known as subjective pure tones. Thus for every pure tone defined by actual frequency

measured in Hz, a subjective pitch is measured on a scale called the mel or bark scale. As a standard reference, a pitch of a 1 kHz tone, 40 dB above the hearing threshold, is defined as 1000 mels. Mathematically it has been shown that the subjective pitch in mels increases less and less rapidly as the stimulus frequency is increased linearly [4][44]].

These perceptual nonlinearities have led to modeling the peripheral auditory system by critical-band filters. The model postulates that sounds are preprocessed by a band of triangular bandpass filters, with center frequency spacings and bandwidths increasing with frequency (equivalently increasing by a constant mel frequency interval) [4]. In fact, these filters are designed similar in spacing as the auditory neurons located on the basilar membrane in the inner ear. The modified spectrum of the speech signal $S(\omega)$ thus consists of the output power of these filters when $S(\omega)$ is the input. If the power coefficients are denoted by $\tilde{S}_k$, $k = 1, 2, \ldots, K$, we can calculate what is called the mel-frequency cepstral coefficients (MFCC) [45] denoted by $\tilde{c}_n$, which can be expressed as

$$\tilde{c}_n = \sum_{k=1}^{K} \log(\tilde{S}_k) \cos[n(k - \frac{1}{2})\frac{\pi}{K}], \qquad n = 1, 2, \cdots, L \tag{2.32}$$

where L is the number of cepstral coefficients. The MFCC has been used for closed set speaker count determination [62] and we compare it to the LPCC and PPF in both the closed and open set situations.

2.5 Pitch

As discussed in section on speech production. the spectral content of the speech signal changes constantly depending on the articulation of the different sounds. Within voiced portions of the speech signal. the signal exhibits a periodic structure which is described as the pitch of a signal. The pitch or fundamental frequency is defined as the interval between glottal pulses and is controlled by the tension of the vocal cords and the buildup of air pressure in the lungs. The glottal pulses are not uniform in terms of their amplitude (volume). and thus the resulting speech signal is amplitude modulated. Pitch variations in the range of $2 - 10\%$ may occur between two consecutive pitch periods [57]. The pitch period feature of the speech signal is discussed in detail in later chapters 4.

2.6 Pattern Recognition Techniques

1 Vector Quantizing Classifier

A vector quantizer (VQ) of size N is defined mathematically as a mapping from the vector space R^k(over the field of real numbers) into a finite set C containing N output points. C is called the codebook and N is called the codebook size. Entries of C, $x_1, x_2, ...x_N$ are called codebooks or codevectors.

A vector quantizer is completely specified by p, C and a set of disjoint regions in R^p which dictate the actual mapping. Suppose C has N entries $y_1, y_2, ...y_N$. For each codevector y_i, there exists a region R_i such that any input vector $x \in R_i$ gets mapped or quantized to i. This region R_i is called a Voronoi region [10] [11] and defined to be the set of all $x \in R^p$ that are quantized to y_i. The properties of Voronoi regions are as follows

1. Voronoi regions are convex subsets of R^p.
2. $\bigcup_{i=1}^{N} R_i = R^p$
3. $R_i \cap R_j$ is the null set for $i \neq j$.

It is seen that the quantizer mapping is nonlinear and many to one and hence non-invertible.

An important ingredient in development of a Vector quanization algorithm is the distortion or distance measure between two vectors $x \in R^p$ and $y =\in R^p$. Most distortion measures satisfy three properties given by

1. Positivity: $d([x], [y])$ is a real number greater than or equal to zero with equality if and only if $[x] = [y]$
2. Symmetry: $d([x], [y]) = d([y], [x])$
3. Triangle Inequality: $d([x], [z]) \leq d([x], [y]) + d([y], [z])$

To qualify as a valid measure for quantizer design, only the property of positivity needs to be satisfied. The choice of a distance measure is dictated by the specific application and computational considerations. We continue by giving some examples of distortion measures.

The L_n distance is given by

$$d([x],[y]) = \sum_{i=1}^{p} |x_i - y_i|^n \tag{2.33}$$

This is a computationally simple measure to evaluate. The three properties of positivity, symmetry and the triangle inequality are satisfied. When $n = 2$, the squared Euclidean distance emerges and is very often used in quantizer design. When $n = 1$, we get the absolute distance. If $n = \infty$, it can be shown that [9]

$$\lim_{n\to\infty} d([x],[y])^{1/n} = \max_i |x_i - y_i| \tag{2.34}$$

There are two necessary conditions for a vector quantizer to be optimal [9] [10]. As before, the codebook C has N entries $y_1, y_2, \ldots\ldots y_N$ and each codevector y_i is associated with a Voronoi region R_i. The first condition known as the nearest neighbor rule states that a quantizer maps any input vector x to the codevector closest to it. Mathematically speaking, x is mapped to y_i if and only if $d([x],[y_i]) \leq d([x],[y_j]) \forall j \neq i$. This enables us to more precisely define a Voronoi region as

$$R_i = \{[x] \in R^p : d([x],[y_i]) \leq d([x],[y_j])\ \forall j \neq i\} \tag{2.35}$$

The second condition specifies the calculation of the codevector y_i given a Voronoi region R_i. The codevector y_i is computed to minimize the average distortion in R_i which is denoted by D_i where

$$D_i = E[d([x], [y_i]) | x \in R_i] \tag{2.36}$$

All the concepts of vector quantization design have been incorporated into the the Linde-Buzo-Gray (LBG) algorithm that is applied as a classifier. The input to the Linde-Buzo-Gray (LBG) algorithm [34] is a training set $T = x_1, x_2, ... x_M \in R^p$ having M vectors, a distance measure, $d([x], [y])$ and the desired size of the codebook N. From these inputs, the codewords y_i are iteratively calculated. The probability density $p(x)$ is not explicitly considered and the training set serves as an empirical estimate of $p(x)$. The Voronoi regions are now expressed as

$$R_i = \{[x_k] \in T : d([x_k], [y_i]) \leq d([x_k], [y_j]) \ \forall j \neq i\} \tag{2.37}$$

Once the vectors in R_i are known, the corresponding codevector $[y_i]$ is found so as to minimize the average distortion in R_i as given by

$$D_i = \frac{1}{M_i} \sum_{[x_k] \in R_i} d([x_k], [y_i]) \tag{2.38}$$

where M_i is the number of vectors in R_i. In terms of D_i, the overall average distortion

D is

$$D = \sum_{i=1}^{N} \frac{M_i}{M} D_i \tag{2.39}$$

Explicit expressions for y_i depend on $d([x], [y_i])$ and two examples are given. For the L_1 distance.

$$[y_i] = \text{median}[x_k \in R_i] \tag{2.40}$$

For the L_2 distance.

$$[y_i] = \frac{1}{M_i} \sum_{[x_k] \in R_i} [x_k] \tag{2.41}$$

which is merely the average of the training vectors in R_i.

The overall methodology to get a codebook of size N is

1. Start with an initial codebook and compute the resulting average distortion.
2. Find R_i.
3. Solve for $[y_i]$.
4. Compute the resulting average distortion.
5. If the average distortion decreases by a small amount that is less than a given threshold, the design terminates. Otherwise, go back to step 2.

If N is a power of 2 , a growing algorithm starting with a codebook of size 1 is formulated as follows

1. Find codebook of size 1.
2. Find initial codebook of double the size by doing a binary split of each codevector. For a binary split, one codevector is split into two by small perturbations.
3. Invoke the methodology presented earlier of iteratively finding the Voronoi regions and codevectors to get the optimal codebook.
4. If the codebook of the desired size is obtained, the design stops. Otherwise, go back to step 2 in which the codebook size is doubled.

Note that with the growing algorithm, a locally optimal codebook is obtained.

2 Neural Tree Network Classifier

The Neural Tree Network (NTN) classifier is a hierarchical classifier that combines the properties of decision trees and feedforward neural networks [29]. The NTN uses a tree architecture to implement a sequential linear decision strategy [46]. The architecture of the NTN is determined during training. Thus, it is self-organizing. Also, NTN training is supervised in that training data pertaining to different conditions (each having a distinct label) is used. Therefore, each training feature vector has a

label indicating the condition from which it emanates. Each node at every level of the NTN divides the input training vectors into a number of exclusive subsets of the training data. If a set of training data at a particular node is of the same class or condition (has the same label), then that node becomes a leaf. Otherwise, the data is split into several subsets, which become children of this node. This procedure is repeated until all the data is completely uniform at the leaf nodes. The leaf nodes of the NTN partition the feature space into homogeneous subsets, meaning a single class at each leaf node. An illustration of this concept is given in Fig. 2.4. In Fig. 2.4, the training data comes from two classes labelled as 0 and 1. The circles represent nodes and the squares represent leaves. The nodes can be thought of as being hyperplanes that partition the space into exclusive subspaces. These subspaces are further partitioned until a leaf is reached.

The NTN will give a 100 percent performance on the training set. Since test data is always different from the training data, an optimal performance is not necessarily reached for a fully grown NTN due to overtraining [29]. Therefore, we use the strategy of forward pruning (has been used for speaker identification) [29] to avoid overtraining. When implementing forward pruning, the NTN is grown only to a specified number of levels and the nodes at the lowest level are said to be leaves. In this case, the training data for a leaf is not necessarily from the same class. A majority vote is taken and the leaf is assigned the label of the majority. We study the speaker count performance with varying number of levels.

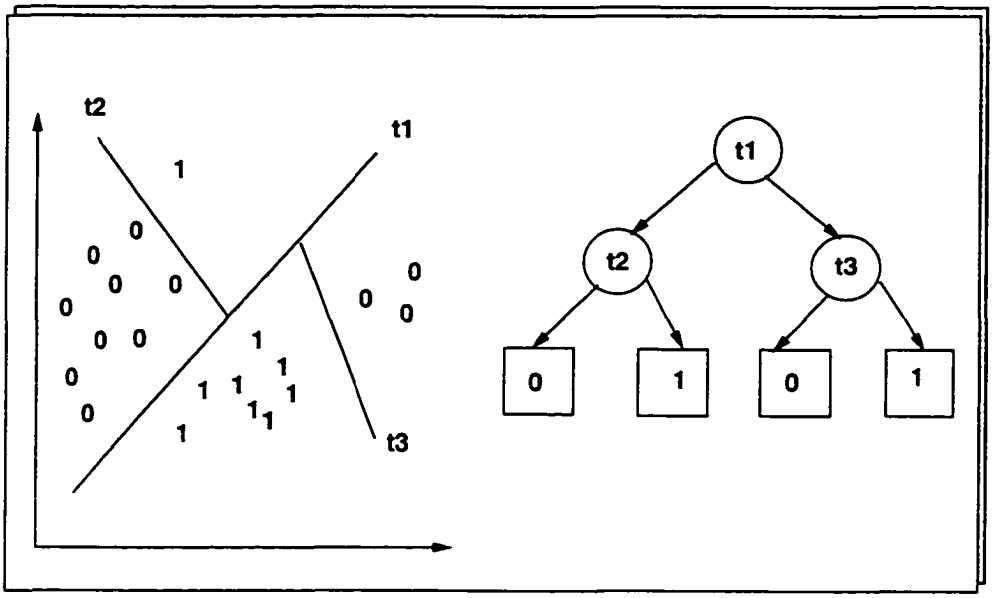


Figure 2.4: Concept of neural tree network. The circles represent nodes and the squares represent leaves.

Chapter 3

Cochannel Speech Labelling Using Pitch Prediction

3 Introduction

Cochannel interference is a situation that develops when a speech signal is corrupted by the voices of other speakers. Though heavily researched in the past two decades, cochannel interference, is still one of the most formidable problems in speech processing. In applications that call for remote access by users, cochannel interference is often the cause of diminished performance. The interference may be introduced in the communication channels, at some point during the transmitting and receiving end. In tactical communication systems, where there are multiple signals transmitted over a

single channel, this problem is also common. It is also possible that interference may be introduced at the transmission site itself. This is often the case if the microphone at the transmitting end is not acoustically isolated, in which case all background noises, including voices would be transmitted along with the primary speaker. This scenario is often exemplified in speakerphones and other hands-free communication devices. No matter where the interference has occurred, the end-result is a corrupted signal of multiple voices that creates major problems for automatic speaker/speech recognition systems.

In this chapter we address the problem of identifying temporal regions of a cochannel signal as being either one-speaker speech or two-speaker speech. This is known as the speaker count labelling problem in that given a temporal region or frame of speech, the label corresponds to a count of either 1 or 2. Solving the speaker count labelling problem is very important in making automatic speaker and speech recognition systems more robust. Suppose that a speaker identification system is trained on one-speaker speech only and a cochannel signal is encountered during testing. There is a mismatch between the training and testing conditions which causes serious performance degradation. If it is possible to label the regions of the cochannel signal that have a count of 1, only the feature vectors from those regions can be used for speaker identification. Similarly, for speech recognition, regions having a count of more than 1 can be further processed to remove the interference. In fact, if the objective is speaker interference suppression, speaker count labelling

can be used in conjunction with a knowledge of the pitch tracks of both speakers to diminish the effect of the interfering speaker.

To the authors' knowledge, the problem of speaker count labelling has rarely been dealt with. In [62], a similar problem known as Automatic Talker Activity labelling has been addressed. In this work, cochannel speech was used as input where frames were labelled either target (primary speaker), jammer (interfering speaker) or talker-jammer (cochannel speech). A classifier was then used to train front-end feature vectors for the 'target' speaker, the 'jammer' speaker and the combination of both speakers. During the recognition, the detector was presented with speech from the target, jammer and combination of target-jammer. The detector's task was to use the stored references to identify which of the three possible sources produced the input and report that result. The detectors were then evaluated on their ability to label the test input correctly. With the use of the mel frequency cepstral coefficient (MFCC) feature and a vector quantizer (VQ) classifier, a 80% correct detection rate was recorded.

The speaker count algorithm we propose is based on a common pattern recognition approach involving feature extraction and classification. We attempt to find features that can discriminate between one-speaker and two-speaker speech on a frame by frame basis. Experiments are conducted with the MFCC feature and the linear predictive cepstral coefficients (LPCC). We also propose a new feature based on the

concept of pitch prediction which is commonly used in speech coding [63]. For the classifier, we compare the VQ [9][37] and the neural tree network (NTN) [40][29]. Two distinctive scenarios are examined, namely, the closed set case where the identity of the speakers is known *apriori* and the open set case where the identity of the speakers is not known. Note that in [62], the closed set case is mostly examined. The speech is assumed to be text-independent in that there is no restriction on what phonemes are uttered.

The outline of this chapter is as follows. Section 2 discusses the various features we use for speaker count. In Section 3, we describe the VQ and NTN classifiers that we along with the features to do speaker count determination as a pattern recognition task. Section 4 gives the experimental protocol and Section 5 discusses the results. In Section 6, we present the conclusions.

4 Features for Speaker Count Determination

In this section, we discuss the various features that are considered to determine the speaker count.

Based on the LP coefficients a_k, it is possible to derive a host of more robust representations of speech. These are the reflection coefficients, log-area ratios, linear prediction cepstral coefficients (LPCC) and the line spectral frequencies (LSF) [42]

discussed in detail in Chapter 2. Recall that the speech can be modeled using the equation

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + r(n) \tag{3.1}$$

where $s(n)$ is the speech signal, $r(n)$ is the error or LP residual, a_k are the LP weights applied to the previous speech samples in estimating the current sample and p is the LP order. The residual signal $r(n)$ is obtained by applying a nonrecursive filter $A(z)$ to the speech as given by

$$A(z) = 1 - F(z) = 1 - \sum_{k=1}^{p} a_k z^{-k} \tag{3.2}$$

We use the LPCC feature for speaker count which, for a minimum phase $A(z)$, can be recursively computed from the LP coefficients by the relationship

$$c_n = a_n + \sum_{k=1}^{n-1} (\frac{k}{n}) c_k a_{n-k}, \qquad 1 \leq n \leq p \tag{3.3}$$

where c_n are the cepstral coefficients. The LPCC feature is commonly used in speaker recognition systems [43] and it is our intention to examine it for speaker count.

4.1 Pitch Prediction Feature (PPF)

Once the speech signal $s(n)$ has been filtered by $A(z)$ (see Eqs. (3.1) and (3.2)), a residual signal, $r(n)$ that is free of near-sample correlations is produced. This

residual signal contains only the distant sample or pitch information. The pitch prediction filter removes the pitch information. In speech coding, pitch prediction is used to parameterize the pitch information which is transmitted along with the LP parameters [63]. The simplest form of the pitch prediction filter has one tap whose transfer function is given by

$$P(z) = \beta_1 z^{-M} \tag{3.4}$$

where the integral delay M represents the pitch period. Since the sampling frequency is unrelated to the pitch period, the individual samples do not show a high period to period distant sample correlation. Therefore, a 3 tap predictor serves like an interpolation filter and provides for interpolated estimates that show higher period to period correlation. The transfer function is

$$P(z) = \beta_1 z^{-M+1} + \beta_2 z^{-M} + \beta_3 z^{-M-1} \tag{3.5}$$

In computing the predictor coefficients and M, consider the situation of a signal that is passed through the prediction error filter $1 - P(z)$ to generate a residual $e(n)$. Assuming a given value of M, the coefficients of $P(z)$ are chosen to minimize the mean-squared residual

$$E_{mse} = \sum_{n=1}^{N} e^2(n) \tag{3.6}$$

where

$$e(n) = r(n) - \beta_1 r(n - M + 1) - \beta_2 r(n - M) - \beta_3 r(n - M - 1) \tag{3.7}$$

and N is the number of samples in one frame. The minimization of E_{mse} leads to a system of equations which can be written in matrix form as $\mathbf{Ac} = \mathbf{d}$. For a 3 tap predictor, the entries of the matrix $\mathbf{A}$ are

$$A(i, j) = \phi(M + i, M + j) = \sum_{n=1}^{N} r(n - M - i) r(n - M - j) \tag{3.8}$$

for $-1 \leq i, j \leq 1$. The vector

$$\mathbf{c} = [\beta_1 \quad \beta_2 \quad \beta_3]^T \tag{3.9}$$

and the vector

$$\mathbf{d} = [\phi(0, M - 1) \quad \phi(0, M) \quad \phi(0, M + 1)]^T \tag{3.10}$$

Specifically, for the one tap case, $\beta_1 = \phi(0, M)/\phi(M, M)$. In order to determine the optimum lag value M, the mean-squared error is minimized by solving the above equations. The resulting error E_{res} is

$$E_{res} = \phi(0, 0) - \mathbf{c}^T \mathbf{d} \tag{3.11}$$

in which the second term in the above equation is a function of M. The optimal value of M is that which maximizes $\mathbf{c}^T\mathbf{d}$. The procedure is to do an exhaustive search of all integral values of M within an allowable range (we used 20 to 147 for the 8 kHz sampling rate) to find the optimal value. Assuming that the off-diagonal terms of $\mathbf{A}$, which represent the near-sample redundancies, can be neglected, the function $\mathbf{c}^T\mathbf{d}$ can be approximately given by [63]

$$\mathbf{c}^T\mathbf{d} \simeq \sum_{m=M-1}^{M+1} \frac{\phi^2(0,m)}{\phi(m,m)} \tag{3.12}$$

Based on these pitch prediction concepts, a new feature for speaker count has been developed. The pitch prediction feature (PPF) is defined as the standard deviation of the differences between the local peaks of the quantity $\mathbf{c}^T\mathbf{d}$ as determined by the pitch prediction method. The local peaks are those peaks of $\mathbf{c}^T\mathbf{d}$ that are above a given threshold. Based on our observations, peaks that are greater than 50% of the global maximum have been chosen as possible pitch peaks. If a frame of a cochannel speech signal has one speaker, strong peaks will occur at multiples of the pitch period. Therefore, the standard deviation of the differences of the peaks will be very small. In Fig. 3.1(a) and (b), a plot of $\mathbf{c}^T\mathbf{d}$ is given for a frame of speech of two different speakers, one with pitch period 35 samples and the other with period 55 samples. When the speech of these two speakers are mixed as a cochannel signal, there will be a considerably larger number of strong peaks of $\mathbf{c}^T\mathbf{d}$. This is due to the strong cross-correlation values between the pitch pulses of the two speakers. For this

reason, the standard deviation of the differences of the peaks will be much higher. Figure 3.1(c) shows the plot of $\mathbf{c}^T\mathbf{d}$ for a frame of cochannel speech.

In Fig. 3.1(a), the peaks of $\mathbf{c}^T\mathbf{d}$ that are above the 50% threshold correspond to 35, 70, 105 and 139 samples (multiples of the pitch period of 35 samples). The differences in the peaks are 35, 35, 35 and 34 samples. The standard deviation of these differences is 0.5 which in turn is the PPF value. In Fig. 3.1(b), the peaks of $\mathbf{c}^T\mathbf{d}$ are 55 and 110. The differences are 55 and 55 which in turn give rise to a PPF value of 0.0. Figure 3.1(c) represents the peaks of $\mathbf{c}^T\mathbf{d}$ for cochannel speech. The peaks are at 35, 78, 105, 110 and 139 samples. The differences are 35, 43, 27, 5 and 29. The PPF value is 14.18.

5 Classifiers

In speaker recognition systems, vector quantizers (VQ) and neural tree network (NTN) classifiers have been used successfully to render decisions about the identity of a speaker [37][29] among a group of M speakers. Each speaker is represented by a VQ codebook or NTN model that is configured during training. During testing, the feature vectors are obtained from one utterance consisting of many frames. These feature vectors are applied to each of the VQ codebook or NTN models (depending on which classifier is used) to get M distinct scores. The model with the best score identifies the speaker. The speaker count determination problem is slightly different

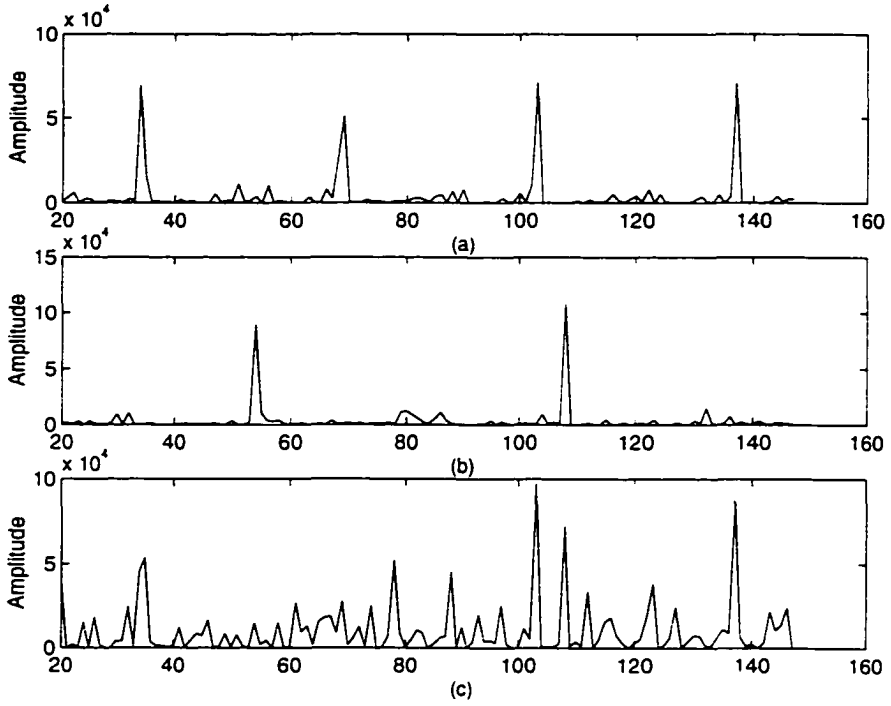


Figure 3.1: Plot of $\mathbf{c}^T\mathbf{d}$ for a frame of a) Speaker 1 with PPF value of 0.5, b) Speaker 2 with PPF value of 0.0 and c) Cochannel signal with PPF value of 14.18

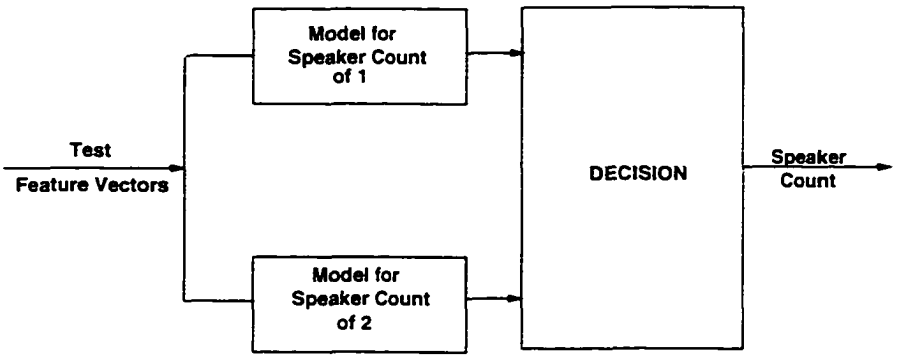


Figure 3.2: Speaker count determination system

in that a model is needed to represent a speaker count of 1 and 2. Also, in contrast to speaker recognition, a decision is taken for each individual frame rather than for an entire utterance. The speaker count is determined for each frame and hence, the decision is taken using only one feature vector. In speaker recognition, an entire utterance is processed and hence, the decision is taken using an ensemble of feature vectors. The general scheme for speaker count is shown in Fig. 3.2.

5.1 Vector Quantizer

In this chapter, two scenarios are investigated for the speaker count determination problem. The first looks at the closed set case, where the speakers are known *apriori*. In this instance, three codebooks are developed from the training feature vectors, each dedicated to one of the three types of possible speech conditions encountered. The three conditions are (1) one-speaker speech from the first speaker, (2) one-

speaker speech from the second speaker and (3) two-speaker or cochannel speech from both speakers. The codebook for each condition is designed by the Linde-Buzo-Gray (LBG) algorithm [64] from training data for that particular condition only. This is known as unsupervised learning in that training data pertaining to another condition does not influence the codebook for a particular condition. Each of the codebooks will have the same size or number of codevectors. We evaluate the performance for various codebook sizes. During testing, consider a test feature vector from a particular frame. It is quantized by each of the three codebooks. The quantized vector is that which is closest according to some distance measure to the test feature vector. We use the squared Euclidean or L_2 distance in our work. Hence, 3 different distances are recorded, one for each codebook. The codebook which renders the smallest distance identifies the speech condition. If condition (1) or (2) results, we have a speaker count of 1. If condition (3) results, the speaker count is 2. The performance is the number of frames identified correctly divided by the total number of frames tested. The next section gives details on how the training data, test data and correct speaker count are obtained.

In the open set case, the speakers are not known *apriori*. Two codebooks are designed by the LBG method, one for one-speaker speech and the other for two-speaker or cochannel speech. The testing is done as in the closed set case but only 2 distances are recorded. The codebook which renders the smaller distance identifies the speaker count.

5.2 Neural Tree Network

The NTN classifier, discussed in detail in Chapter 2, is a hierarchical classifier that combines the properties of decision trees and feedforward neural networks [29]. For the speaker count determination closed set problem, an NTN is grown from training data consisting of three labels. The three labels are for the three conditions which as before are (1) one-speaker speech from the first speaker, (2) one-speaker speech from the second speaker and (3) two-speaker or cochannel speech from both speakers. Given a frame of test speech, the feature vector is found and passed through the NTN so that is reaches a particular leaf. The label assigned to the leaf classifies the speech frame. For the open set case, an NTN is grown from training data consisting of two labels, namely, one-speaker and two-speaker speech. The classification of a test feature vector is similarly done in that a match is made to the label of the leaf reached.

6 Experimental Protocol

In the following section, the experimental protocols for the closed and open set schemes are discussed. In general, the experimental protocols are quite similar. However, there are some slight variations. A key point to note, is that, in the closed set experiment, particular attention is paid to the speakers' identity. In the open set case, however, the identity of the speakers is not relevant. We will first describe the

feature computation and then look into the training and testing phases. The New England portion of the TIMIT database is used in which the speech is downsampled to 8 kHz.

6.1 Feature Computation

The computation of the features applies to both the training and testing phases. For the LPCC feature, the autocorrelation method of LP analysis is used to get the LP coefficients a_k [41]. The speech signal is first preemphasized by passing it through a nonrecursive filter $1 - 0.95z^{-1}$. A 30 ms long Hamming window is then applied with a 20 ms overlap thereby providing a frame size of 10 ms. A 12th order LP analysis is done and the LP coefficients a_k are converted into a 12th order LPCC feature vector using the recursion in Eq. (3.3). A 12 dimensional MFCC feature is also calculated. As for the LPCC feature, preemphasis followed by a 30 ms Hamming window with a 20 ms overlap is applied.

Consider the PPF feature. In this case, no preemphasis is applied. A 12th order LP analysis by the autocorrelation method using a 30 ms Hamming window with a 20 ms overlap gives us the LP coefficients a_k. Using the a_k, the speech is filtered to generate the LP residual as in Eq. (3.1). Prior to extracting the PPF, the LP residual is passed through a Gaussian shaped filter whose impulse response $f(\frac{x}{\sigma})$ can

be expressed as

$$f(\frac{x}{\sigma}) = \frac{1}{\sigma\sqrt{\pi}} \exp(-\frac{x^2}{2\sigma^2})$$

The impulse response has a Gaussian shape and σ refers to the standard deviation of the Gaussian function. This type of filter has been used in image processing, particularly for edge detection [47][48]. The utilization of this filter is motivated by our observation that it acts so as to smooth the LP residual thereby enhancing the performance of the peak picking algorithm (to pick the peaks of $\mathbf{c}^T\mathbf{d}$ as described earlier) used when generating the PPF. Different values of σ were tried and the best performance was achieved with a $\sigma = 0.32$. The number of filter taps is equal to an odd integer closest to $16(8\sigma + 1)$. From the Gaussian filtered LP residual, the pitch prediction algorithm was applied to get a 3 tap pitch filter $P(z)$ and the quantity $\mathbf{c}^T\mathbf{d}$ for $M = 20$ to 147. For the pitch prediction algorithm, a framesize of 10 ms was used (performance given later) and there was no overlap between frames. After finding the global maximum of $\mathbf{c}^T\mathbf{d}$, a threshold equal to 50% of this maximum was set. Again, different thresholds were tried before a decision was taken. The local peaks of $\mathbf{c}^T\mathbf{d}$ are those above the threshold from which the candidate values of M are taken. From these candidate values of M, the PPF scalar feature is found by taking the standard deviation of the differences as described earlier.

6.2 Training Phase

During the training phase for all the experiments, the general aim is to derive features that represent one-speaker and two-speaker or cochannel speech. The first five sentences for each of the 38 speakers in the New England portion of the TIMIT database represent the training speech. Six speakers (3 male and 3 female) are selected and all possible sentence combinations are used to derive the training cochannel speech. With this exhaustive combining method, a total of 375 cochannel sentences are used for training. In generating the cochannel sentences, the individual sentences are first normalized by their maximum absolute sample value before being added. In explaining both the closed and open set cases, let the individual speech signals pertaining to speaker A and speaker B be $s_A(n)$ and $s_B(n)$ respectively. The cochannel signal is denoted as $s_{AB}(n) = s_A(n) + s_B(n)$.

Consider the closed set scenario. The signal $s_A(n)$ is divided into frames and energy thresholding is used to distinguish between speech frames and silent frames. For the speech frames, a simple voiced/unvoiced detector establishes the voiced frames of speaker A. This simple detector is based on locating the peak of the magnitude of the LP spectrum. If this peak occurs at a frequency at or below 2 kHz, the frame is said to be voiced. Note that the voiced/unvoiced detector is very important in improving the performance of the PPF feature. The same procedure is repeated for $s_B(n)$ to get the voiced frames of speaker B. For the cochannel signal $s_{AB}(n)$,

the cochannel frames are those for which a voiced frame of speaker A and a voiced frame of speaker B are added. The frames of $s_{AB}(n)$ for which a voiced frame of speaker A and a silent or unvoiced frame of speaker B are added correspond to a voiced frame of speaker A only. Similarly, when a voiced frame of speaker B and a silent or unvoiced frame of speaker A are added, we get a voiced frame of speaker B only. We now gather voiced frames of speaker A from $s_A(n)$ and $s_{AB}(n)$, voiced frames of speaker B from $s_B(n)$ and $s_{AB}(n)$ and cochannel frames of both speakers from $s_{AB}(n)$. The features are computed for these three cases and the VQ and NTN classifiers designed.

Consider the open set scenario. As in the closed set case, energy thresholding and voiced/unvoiced detection are performed on $s_A(n)$ and $s_B(n)$ to get the voiced frames. These voiced frames are one-speaker frames. For the cochannel signal $s_{AB}(n)$, the two-speaker or cochannel frames are those for which a voiced frame of speaker A and a voiced frame of speaker B are added. From $s_{AB}(n)$, we also extract one-speaker frames when a voiced frame of one of the speakers is added with a silent or unvoiced frame of the other speaker. The features are computed for the one-speaker and two-speaker cases and the VQ and NTN classifiers designed.

6.3 Testing Phase

In the testing phase, energy thresholding and voiced/unvoiced detection are performed on the cochannel signal $s_{AB}(n)$. For the frames which are declared to be voiced, the feature is computed and classified by either the VQ or NTN classifier as described above. For the closed set case, the decision is one-speaker speech from speaker A, one-speaker speech from speaker B or two-speaker speech. For the open set case, the decision is either one-speaker or two-speaker speech. To do the testing, six speakers from the TIMIT database that are different from those used for training are chosen. Three of the speakers are male and three are female. There are five testing sentences for each speaker that are different from the sentences used for training. All possible sentence combinations are used to derive the 375 cochannel speech sentences used for testing.

In measuring the performance, the decision must be compared to some notion of a correct answer which is not as obvious as in the case of assessing speaker identification systems. We formulate one simple method to get a correct answer as follows. Given a cochannel signal $s_{AB}(n)$, energy thresholding and voiced/unvoiced detection are performed on the constituent signals $s_A(n)$ and $s_B(n)$ as is done during training. The rest of the training procedure is essentially repeated to label the cochannel frames. The performance is the number of times a frame is classified correctly divided by the total number of frames tested (82,174 in our experiments). There are two sources

of error given a particular cochannel frame. The first is when a decision is taken but does not correspond to the correct frame label. The second is when a decision is taken (since the cochannel frame is declared to be voiced) but there is no frame label (since neither the corresponding frame of $s_A(n)$ and $s_B(n)$ are declared to be voiced). This second source of error is very rare and occurs less than 0.1 percent of the time.

7 Results and Discussion

In this section, the performance of the new PPF and cepstral features are compared. In the first set of results, the performance of the VQ and NTN classifiers are compared in the closed set case. In the second set of results, the experiments are repeated for the open set case. The vector quantizer codebook sizes range from 16 to 256 for the LPCC and MFCC features. Lower codebook sizes of 1 to 16 were used in the case of the PPF. This is due to the fact that the PPF is a scalar feature as compared to the twelve dimensional LPCC and MFCC features. Therefore, the use of higher codebook sizes for a scalar feature is not necessary and actually diminishes the performance. For the NTN classifiers tree levels of 2 to 10 were grown for all the features.

Codebook size	Cepstral Feature	
	LPCC	MFCC
16	83.1	75.6
32	83.2	75.9
64	83.1	76.1
128	83.1	76.2
256	83.1	75.5

Table 3.1: Closed set results for the cepstral features using the VQ classifier

7.1 Closed Set Case

Tables 3.1-3.3 depict all the closed set results. The VQ classifier outperforms the NTN for all the features. We concentrate on the results obtained using VQ. The LPCC feature outperforms the MFCC for all the VQ codebook sizes. The performance of the PPF is essentially equal to that of the LPCC. The PPF still maintains an advantage in that the feature dimension is substantially lower. Moreover, the smallest codebook size of 1 can be used as negligible performance gain is achieved by a larger codebook size.

Codebook size	PPF Feature
1	83.2
2	83.3
4	83.3
8	83.3
16	83.5

Table 3.2: Closed set results for the PPF feature using VQ classifier

Number of Levels	Feature		
	LPCC	MFCC	PPF
2	65.9	63.9	65.2
4	64.8	65.6	66.6
6	64.6	65.2	66.6
8	64.7	64.8	66.9
10	64.5	64.6	65.6

Table 3.3: Closed set results for the cepstral and PPF features using the NTN classifier.

Codebook size	Cepstral Feature	
	LPCC	MFCC
16	56.8	58.8
32	56.7	58.9
64	56.7	57.7
128	56.7	57.5
256	56.8	57.3

Table 3.4: Open set results for the cepstral coefficients using the VQ classifier.

7.2 Open Set Case

Tables 3.4-3.6 depict all the open set results. Since the open set case is a harder problem than the closed set case, the performance is less for the open set case. The VQ classifier again essentially outperforms the NTN. We concentrate on the results obtained using VQ. The LPCC and MFCC show a similar performance. The PPF shows the best performance for the smallest codebook size of 1 and outperforms the cepstral features.

Codebook size	PPF Feature
1	64.4
2	63.0
4	50.2
8	55.0
16	61.1

Table 3.5: Open set results for the PPF feature using the VQ classifier

Number of Levels	Feature		
	LPCC	MFCC	PPF
2	55.8	49.0	50.5
4	54.4	60.2	48.8
6	52.7	55.8	47.0
8	51.6	47.5	51.2
10	48.4	50.7	51.2

Table 3.6: Open set results for the cepstral and PPF features using the NTN classifier

Codebook size	LP Feature			
	Reflection Coefficients	Log-area Ratios	Area Coefficients	Line Spectral Frequencies
16	52	53	55	53
32	52	55	53	52
64	56	55	53	53
128	55	53	54	53
256	55	54	55	53

Table 3.7: LP Features Accuracy for Speaker Count (134,882 frames).

3.1 Comparison of PPF with other LP features

Several other LP features discussed in Chapter 2 were also tested under similar conditions to the MFCC and LPCC features for the openset case using the VQ classifier. The PPF feature clearly outperforms all these LP features as shown in Table 3.7.

The best results for the LP features are obtained with a modest codebook size of 64. Although the reflection coefficients and the log-area ratios give the best results amoung these LP features, all show an approximately equivalent performance.

1 Summary and Conclusions

We examine the problem of identifying temporal regions or frames of cochannel speech as being either one-speaker or two-speaker speech. Ideally, separation of the individual speech signals that form a cochannel signal is desired. However, it is known that when two equal bandwidth signals are added, such a separation is not possible. Identifying frames as being one-speaker or two-speaker speech is done using a pattern recognition framework based on feature extraction and subsequent classification. We develop a new feature called the pitch prediction feature (PPF) based on the concept of pitch prediction that is used in speech coding. The PPF is a scalar feature that outperforms the linear predictive cepstrum (LPCC) and the mel-warped cepstrum (MPCC) both of which are 12 dimensional vector features. The vector quantizer (VQ) and neural tree network (NTN) classifiers are compared and the VQ is found to be consistently better. Note that the superiority of the PPF is not only synergistic with achieving a lower feature dimension but also with being able to use a lower VQ codebook size. In fact, the lowest codebook size of 1 is used for the PPF which essentially is equivalent to a Bayesian discriminant approach [46]. Two cochannel scenarios are looked at, namely, the case when the speaker identities are known apriori (closed set) and when the identities are not known (open set). The open set problem is more difficult and as expected, the performance for all the features is less. For both the open set and closed set problems, the PPF is the best feature. In addition, the PPF outperformed all the other LP features in the openset

case with the VQ classifier. In Chapter 6, an algorithm is proposed to use speaker count labelling to enhance speaker ID results.

Chapter 4

Robust Pitch Estimation Methods

4.1 Introduction

Pitch is one of the most important parameters in the analysis and synthesis of speech. Accurate and robust pitch determination, however, is clearly one of the more difficult tasks to achieve in speech processing. The difficulty arises from the irregular and variable nature of the speech signal. Firstly, the human vocal tract varies tremendously from person to person. In fact, the pitch period of humans can vary from 3 ms to 40 ms [13]. Secondly, the pitch period can vary depending on the emotional state, accents and other perceptual variables [14]. Thirdly, in telephony and other applications, the pitch period of the signal can be affected by noise, phase distortion or bandwidth reduction of the signal [13]. Therefore, developing an algorithm that

can perform well for different speakers (e.g. male, female or people of different languages, etc.), for different applications and under different environmental conditions is greatly needed.

The pitch period has been used in many applications over the years. In vocoders, accurate pitch period determination is essential for the preservation of the quality of speech signals since the human ear is very sensitive to changes in pitch [13]. In speaker verification, speaker identification, and speech recognition systems the pitch period detection and extraction is necessary [15]. In the field of medicine, particularly phoniatry, the pitch period is used in the early detection of voice diseases [53]. Pitch period estimation is a also an excellent pre-processing block for speech enhancement systems using comb filtering [38]. An accurate pitch estimate leads to an accurate comb filter and successful removal of the noise in the speech signal.

The pitch period is produced in voiced signals when the vocal cord is tensed and therefore vibrates periodically when air flows from the lungs. The resulting voiced speech waveform can be considered periodic or quasi-periodic. In voiced signals, the cycle of vocal cord vibration consists of two events - the glottal opening and glottal closing. The time interval between the two successive glottal closures, is often defined as a pitch period.

Over the years there have been several Pitch Detection Algorithms (PDA) proposed to determine pitch period. These techniques can be classified into three main cat-

egories: time domain, frequency domain and event-based techniques. In the time-domain category several pitch period schemes have been developed [18] [20]. These measurements are reasonably accurate, however, these algorithms have not been shown to be very robust to noise and work poorly for different types of speakers. In addition, these methods only estimate the average pitch periods of the speech signal in an analysis window.

The frequency domain detectors make use of the property that if the signal is periodic in the time domain, then the spectrum of the signal will consist of a series of impulses at the fundamental frequency and its harmonics Noll [21]. The advantage of the frequency domain detector is that it is less susceptible to noise. It can also eliminate or reduce the effect of the formant frequencies on the pitch period. These methods also assume stationarity within the speech signal. As a result, these detectors are insensitive to non-stationary variations in the pitch period over the segment length. Another disadvantage, is that these algorithms are unsuitable for use on both high and low pitched speakers.

Event based pitch detectors depend on locating the instant at which the glottis closes or points of sharp discontinuities in the speech signal. As previously defined, the pitch period is the distance between two glottal closures. Therefore, an event based pitch detector does not assume quasi-stationary of the speech signal. This means that it does not estimate the average pitch period, but the actual pitch period. The

autocovariance method used by Strube et al.[59] determines the actual pitch period based on this principle. By determining the maximum of the determinant of the autocovariance matrix, the point of glottal closure can be determined. However, this method is good only for specific voiced signals and is computationally intense.

One of the most accurate and promising event-based pitch detectors utilizes the Dyadic Wavelet Transform (DyWT) introduced by Mallat et al. [12]. This algorithm was later adapted by Kadambe et al. [55] for the determination of the pitch period in speech. The algorithm uses the multiresolution property of the DyWT to smooth the signal and filter the noise. In this chapter, we introduce the Adaptive Gaussian Derivative Filter (AGDF) which uses similar principles to adaptively smooth the signal and determine pitch period information under noisy conditions. We compare this adaptive smoothing approach to Kadambe's algorithm to determine both the algorithm's robustness and accuracy under different types of noisy conditions. The motivation for the use of this function stems from recent developments in the field of edge detection in digital images where the Gaussian Derivative function has been used to smooth noise and enhance edges [50]. Based on the definition of event base pitch detection, detecting the sharp discontinuities at the point of glottal closures is the one dimensional case of detecting the edges in digital images. This Gaussian Derivative function which is the summation of the zeroth and second order hermite function, was used by Basu et. al.[50] in edge detection and has been used here in the determination of the pitch period. Based on the results, the adaptive gaussian

derivative filter proves to be extremely accurate and more robust to different types of noise than the DyWT method when applied to synthesized and real speech under various conditions of noise.

4.2 Algorithms

In this section, we describe the basic algorithms used for the DyWT and AGDF algorithms for robust pitch detection.

1 DyWT Pitch Detector

Wavelets are operators that have the ability to be shifted and scaled. The Continuous Wavelet Transform is defined by [32] as a function that has a zero mean and satisfies the form

$$CWT_f(a,b) = \frac{1}{\sqrt{a}} \int_R \psi * \left(\frac{t-b}{a} \right) f(t)dt \tag{4.1}$$

The factor $1/\sqrt{a}$ is used to conserve the norm and $\psi * (t)$ is the complex conjugate of $\psi(t)$. The CWT is seldom used because it is computationally complex and it has a lot of redundant information since it is a two dimensional expansion of a one dimensional function [32]. Instead other forms of Wavelet transform are used, namely the discrete Wavelet transform and the dyadic Wavelet transform (See Appendix B).

For speech and image analysis, the Dyadic Wavelet Transform is most often used. The continuous DyWT of a signal $x(t)$ can be expressed similarly to equation Fig. 4.1. However, in the dyadic case the scaling function is given by $a = 2^j$ where $j = 1, 2, ...$ which means that it is discretized along a dyadic sequence. Mallat [12] considered a Wavelet which is the first derivative of a smoothing function (See Fig. 4.1). The general properties are particularly good for speech analysis for the following reasons. The DyWT has linear and shift invariant properties which are useful since speech is often modeled as a linear combination of shifted and damped sinusoids. Its multi-resolution properties makes the DyWT very attractive since the signals can be examined at different levels of detail. In addition, the modulus of the DyWT of a signal x(t) exhibits local maxima around the point of discontinuity [12]. This is significant in the use of DyWT in speech since at the point of glottal closure there is a sharp discontinuity or transient in the speech signal. In particular, the local maximas of the DyWT indicate the sharp variation in a signal, whereas the local minima indicate the slow variations.

This multi-resolution analysis of a signal can be described as a hierachial system of subspaces of different resolution that are orthogonal in nature, and as a result are uncorrelated. These subspaces are square integrable and one dimensional. Each subspace is spanned by basis functions that have scaling characteristics of either dilation or compression depending on the resolution or scale. This basis function can be represented by characteristic high and low pass filters which are used in a

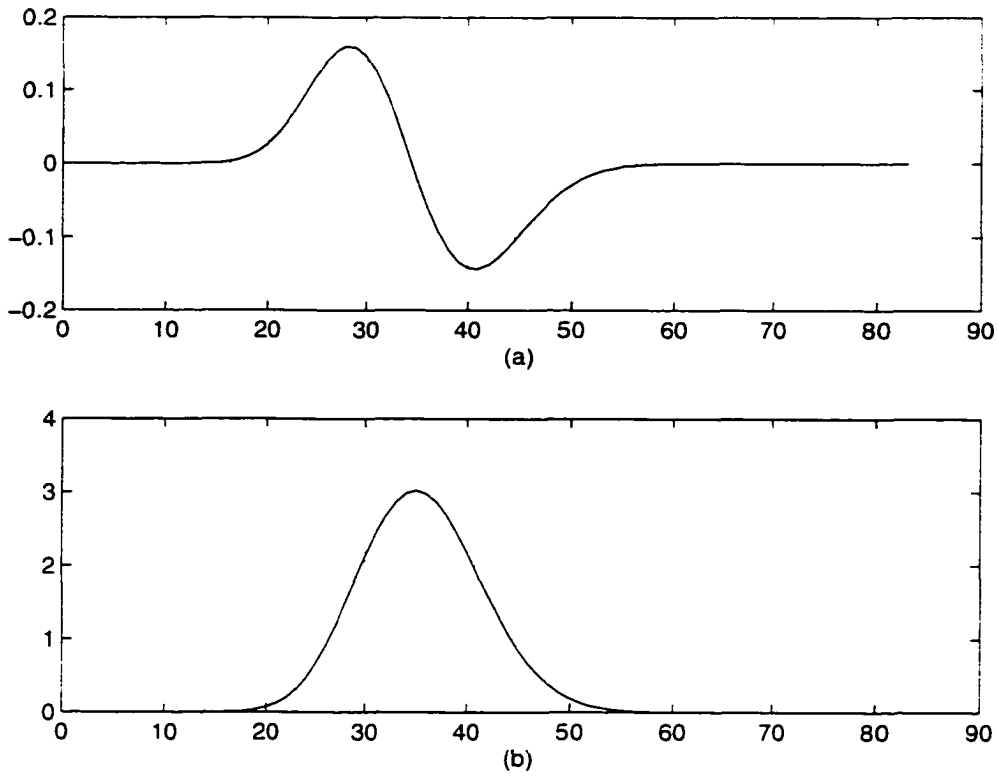


Figure 4.1: a) A quadratic spline or mallat Wavelet which is compactly supported and continuously differentiable. This Wavelet is the first derivative of the function in b)

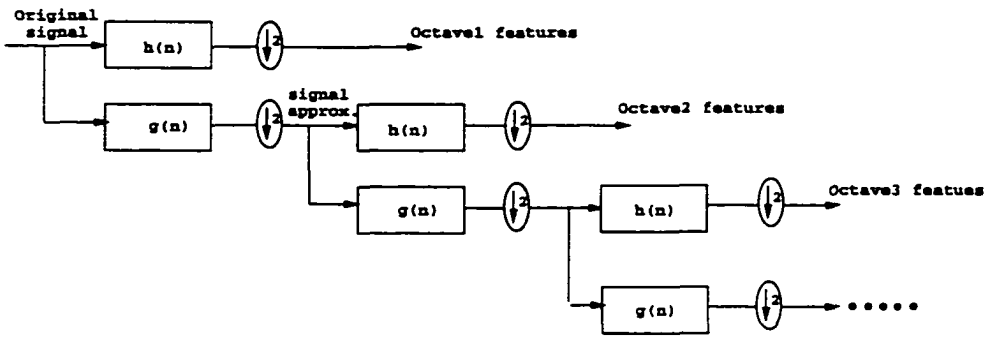


Figure 4.2: The Wavelet filter bank scheme

pyramidal scheme of convolution and down-sampling as illustrated in Fig. 4.2.

The discretized signal is convolved with these low and high pass filters that are complements of each other. The general relation between both filters is $g(n) = (-1)^{(1-n)}h(1-n)$. These filters are designed based on the characteristics of the Wavelet transforms and their scaling functions. The output of these filters are then down-sampled or decimated. In this decimation one can either take every other sample, or insert zeros between every sample of the filter which is equivalent to down-sampling process [12]. This means that the sub-sampled signal would be down-sampled by a factor of two. The output of the high pass filter g(n) is known as octave features and represents the signal detail at a particular resolution. The output of the low pass filter or the approximated signal is then passed into successive stages of low and high pass filters as shown in Fig. 4.2.

The wavelet coefficients are projected onto frequency bands that are contiguous in nature. Thus there are no overlaps in the output frequencies or octaves which results from the convolution and down-sampling at different stages.

Based on these principles, Mallat developed an algorithm for multi-scale edge detection in images [12]. This algorithm was later modified by Kadambe et al. for use in speech [55]. In Kadambe's algorithm, the instant of glottal closure is detected by locating the local maxima of DyWT which exceed the threshold. Kadambe's algorithm considered the threshold to be 80% of the global maxima. The pitch period is then estimated to be the time interval between two such local maxima. The steps of the algorithm are summarized as follows:

- The DyWT is computed at scales $a = 2^j$ for all j. For purposes of pitch estimation $j = 4$ is the first scale on which the DyWT is considered. In fact, two scales are compared at the same time. This means that we find the global maximum value of $DyWT(b, 2^j)$ and $DyWT(b, 2^{j+1})$.

- Locate the maxima present in the DyWT computed at the two scales mentioned above, which exceeds the threshold value.

- Check whether the number of local maxima of the DyWT and their locations match. If the locations match, choose the lower scale and estimate the pitch period. If they do not match, increment j by 1 and

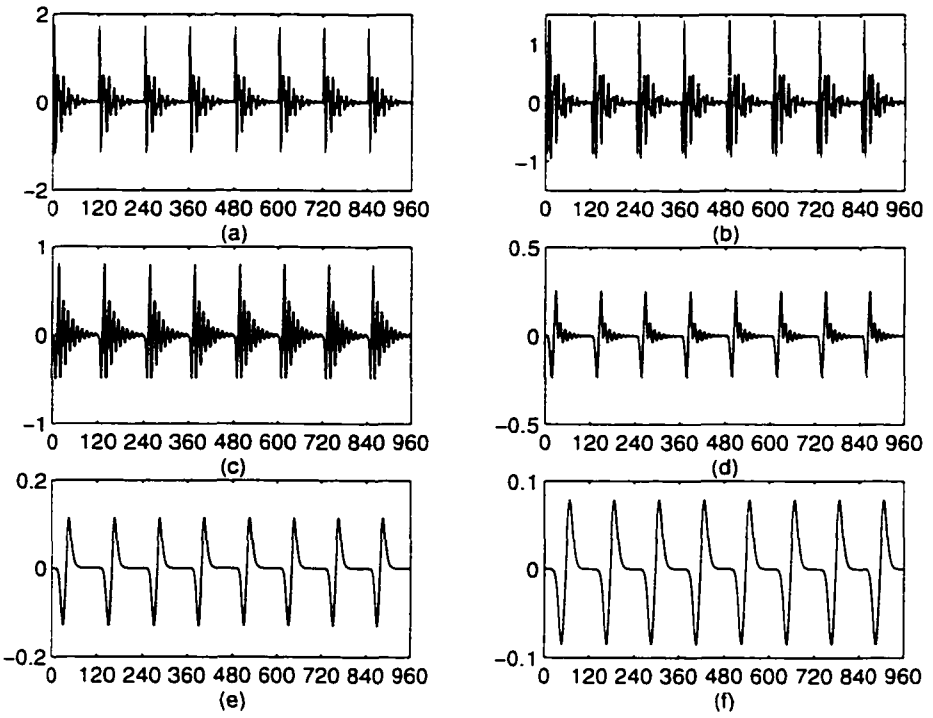


Figure 4.3: a) A synthesized signal /a/(the ticks indicate the onset of the true pitch period). DyWT of /a/ computed at scales b) $a = 2^1$ c) $a = 2^2$ d) $a = 2^3$ e) $a = 2^4$ f) $a = 2^5$. The abscissa shows the number of samples of a signal which is pitch period=15 ms and is sampled at a rate of T=.125 ms.

- Repeat the above steps until $j = 6$. If the number of local maxima and their locations do not match over two consecutive scales from $j = 4, 5, 6$ then the segment of speech is classified as unvoiced. To illustrate these concepts, the DyWT of a synthesized signal is shown over several scales in Fig. 4.3

2 Adaptive Gaussian Derivative Filter

2.1 The Gaussian Derivative Filter

Due to multiple sources of noise, processing of the speech signal is complex and the detection of points of sharp discontinuities or transients are nontrivial. In image processing, several approaches have been studied to obtain the best filter that detect edges and lines within images. In physiology, it is a well known fact that the human visual system extracts features from images by performing detailed edge and line detection. This has motivated many scientists to attempt to model the human visual system response. Based on the study of human physiology, Marr and Hildreth [56] in 1980, introduced the gaussian filter for edge detection. The gaussian filter is known to have good smoothing and localization properties in both the frequency and spatial domain. In fact, the gaussian filter is the only that optimizes the uncertainty principle, which states that $\Delta x \Delta \omega \geq \frac{1}{4}\pi$. Marr and Hildreth found that by convolving the Laplacian of the gaussian with a signal and taking the zero-crossings, it is possible to detect edges in the presence of noise.

In subsequent studies, Young [60] found that the human visual system was more accurately modeled by adding a gaussian term to the Laplacian of the gaussian function. This gaussian derivative filter (GDF) as described by Basu [50] was shown to give better edge enhancement and noise suppression in digital images when compared

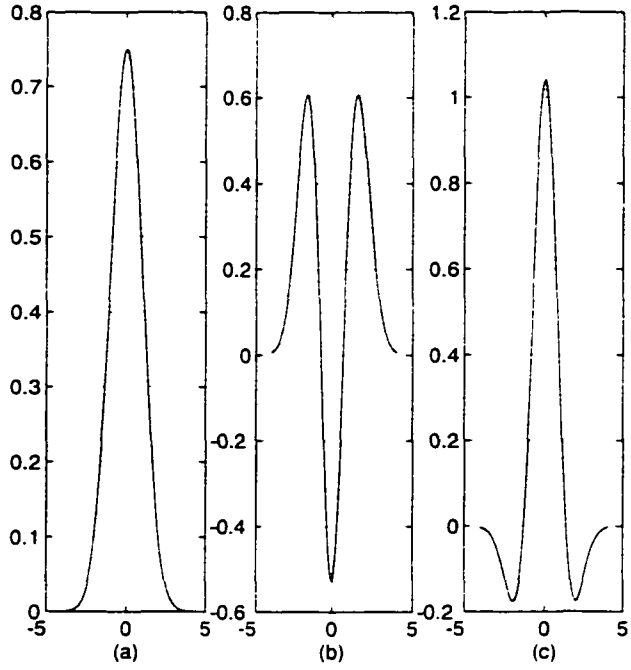


Figure 4.4: a) The zeroth-order hermite function b) The second-order hermite function c) The gaussian derivative function

to the Laplacian of the gaussian filter. Although the utilization of these filters are based on human physiology, the characteristics are well suited for dealing with noise suppression and line enhancement in all types of signals, including speech.

The GDF is the linear combination of Hermite functions (h_n) which are generally described by

$$h_n(x/\sigma) = \frac{1}{\sqrt{2^n n!}} \frac{d^n}{d(x/\sigma)^n} \frac{1}{\sigma\sqrt{\pi}} e^{-x^2/2\sigma^2} \tag{4.2}$$

The 1D-GDF as expressed by [50] can be described by the following equation:

$$g_D(x/\sigma) = c_0 h_0(x/\sigma) + c_2 h_2(x/\sigma) \tag{4.3}$$

where σ is a scaling parameter. Basu [50] found that the coefficient c_0 was responsible for suppressing noise, whereas c_2 maximized the edge magnitude.

Based on equation 4.2, $h_0(x/\sigma)$, the gaussian function that can be described as

$$h_0(x/\sigma) = \frac{1}{\sigma\sqrt{\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{4.4}$$

and

$$h_2(x/\sigma) = \frac{1}{\sqrt{8\pi\sigma^2}}\left(-\exp\left(-\frac{x^2}{2\sigma^2}\right) + \frac{x^2}{\sigma^2}\exp\left(-\frac{x^2}{2\sigma^2}\right)\right) \tag{4.5}$$

In Fig. 4.4, the gaussian derivative function together with the zeroth and second order hermite functions are illustrated. Note that both the zeroth and second order Hermite functions are symmetric and therefore orientation independent operators (See A).

In event based pitch period detection, the differences between consecutive transients or discontinuities created at the point of glottal closure is used to determine the pitch period. When the GDF is convolved with a speech signal, it smoothes or averages the high frequency components in the signal while preserving its slowly

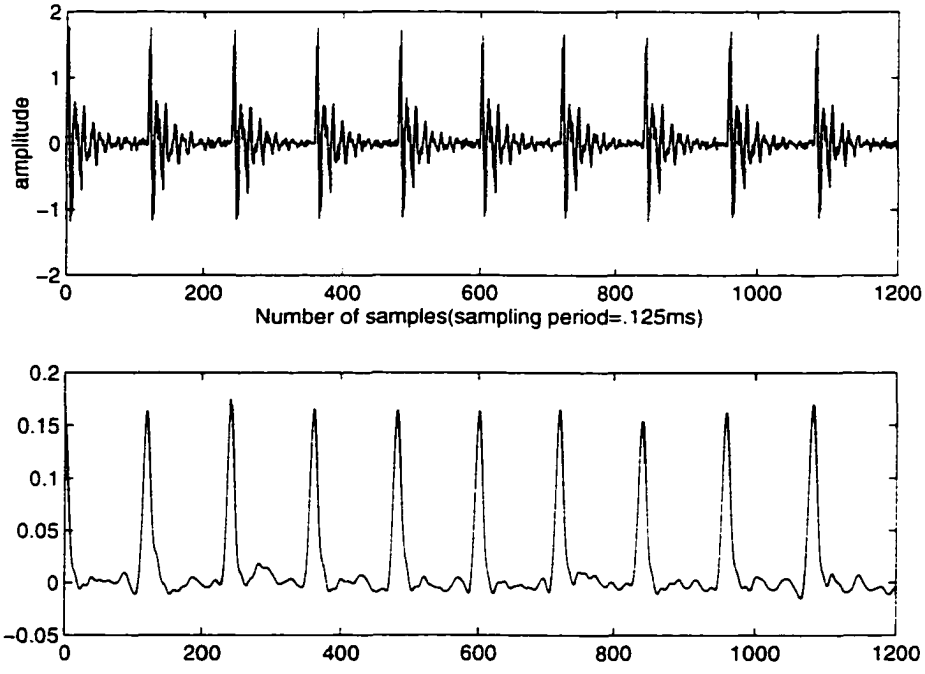


Figure 4.5: a) Synthesized signal /a/ corrupted with $20dB$ noise. b) The effects of applying the GDF on the synthesized signal in a)

varying components. In addition, the transients within the signal are enhanced or made more prominent. Therefore, when a threshold is applied to the result of the GDF convolved with the preprocessed speech, it is less likely to pick up maximas or false positives due to noise. An example of the GDF performed on synthesized speech is shown in Figure 4.5.

The Gaussian derivative operator also provides good localization of the transients within a signal. The aim of any good edge detecting operator is to avoid major

random shifts of the edges in the enhanced speech. This property is particularly important when used for the determination of the pitch period in the speech signal since the aim is to accurately measure the relative positions of the glottal pulses within the signal. In [50], the localization property was demonstrated by comparing the GDF and the Laplacian of the Gaussian(LOG) operator. The effectiveness of the GDF was investigated by looking at the edge localization error for both the GDF and LOG under similar conditions of noise. It was found that in cases of low signal to noise ratio that the probability density function of the edge location error of the GDF was half that of the LOG function. This suggested that the transients in any signal will be better enhanced with the GDF. Also, the position of the transients are less likely to experience major shifts when processed with the GDF.

A major problem associated with using a GDF method for pitch detection, however, is that it does not give the same level of performance for speakers with different pitch, contents of the speech or different level of noise. This is due to the fact that there is no way of controlling the smoothing and the detection of the transients in the signal. In the next two sections, we discuss an Adaptive Gaussian Derivative Filter (AGDF) algorithm to overcome the problems faced by the GDF.

3 Preprocessing

One of the major problems experienced in speech processing is the determination of voiced/unvoiced frames. The problem is further complicated when the speech is in the presence of noise. In many practical cases of pitch detection, the pitch information present in a voiced speech segment of speech is often corrupted by noise. It is important therefore to discriminate between voice/unvoiced frames and voice frames in the presence of noise. The algorithm should robustly reject unvoiced frames while accepting voiced frames even when corrupted by noise. In [67], the problem of frame selection of voice segments of speech was described for speaker identification systems. For speaker identification, the frame selection criterion is based on two main observations. Firstly, voiced speech frames provide most of the discriminative ability for speaker ID. Secondly, speech frames with formant-like spectra are more robust to noise. This second observation is utilized in the AGDF algorithm to correctly discriminate between voiced and unvoiced frames in the presence of noise.

The algorithm in [67] has been slightly modified and used for voice/unvoiced/silence detection for the DyWT and AGDF pitch estimation methods. The algorithm can described as follows:

- A speech signal is windowed into short frames of 240 sample segments with an overlap of 160 samples.

- Silence detection is performed using an energy thresholding method to detect silent frames which is not processed further.

- Preemphasis is applied using a single tap filter with transfer function $H(z) = 1 - .95z^{-1}$

- The frame is LP filtered (refer to Chapter 2) with prediction order of 12 for 8 kHz speech and the roots of the error transfer function $A(z) = 1 - \sum_{k=1}^{p} a_k z^{-k}$ are derived. The roots of the transfer function are directly related to the formant information.

- The LP filter is an all-pole model of speech which has all its poles within the unit circle in order to be stable. The roots in the upper part of the unit circle which have a magnitude greater than .85 and angle which lie between the range of 200 and 3800 Hertz denoted by α in Figure 4.6 is considered. If the number of poles which meet these criteria is greater than or equal to three, the frame is labelled voiced. Otherwise the frame is considered unvoiced.

The algorithm has been tested on an unvoiced and voiced signals in the presence of noise to determine its robustness. In order to illustrate the effectiveness of the silence/voiced/unvoiced detector, the following experiments were performed to illustrate the effectiveness of the detector.

- In Figure 4.7, a frame of the voiced signal 'a' spoken by a male speaker is tested

using the algorithm. Note that there are three roots of the polynomial A(z) which lie outside the radius threshold of .85 and within the angle α.

- In Figure 4.8, there are still three poles outside the .85 radius threshold and angle α even when the frame of voiced signal is corrupted with -5 dB of white gaussian noise.

- In Figure 4.9, the poles of an unvoiced frame of a female speaker uttering 'sh' is examined. Notice there is only one poles within the constraints set by the algorithm.

- In Figure 4.10, the previous unvoiced frame is corrupted with -5 dB of white gaussian noise. Based on the criteria set by the algorithm, the frame is still classified as unvoiced.

These observations are consistent with other types of noise.

3.1 Pitch Detection

The general idea behind adaptive smoothing is to apply a versatile operator which adapts itself to the local topography of the signal to be smoothed. There are several methods of adaptive smoothing approaches discussed in [58] [51] [52]. The simplest and direct smoothing approach uses the basic idea which consists of selecting neighboring points with a central point and replacing the latter by the average of these

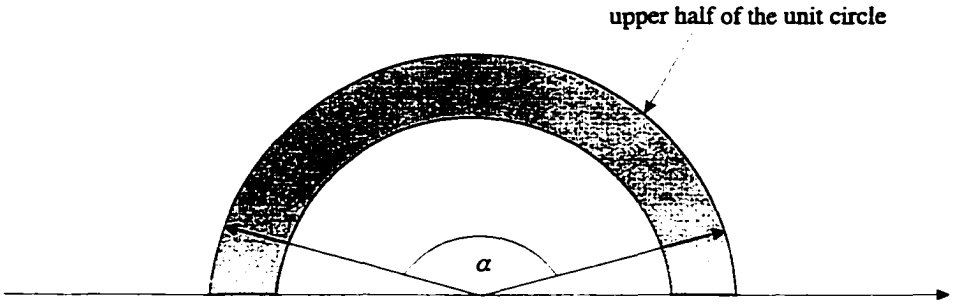


Figure 4.6: The criteria set for voiced frame selection (After [67])

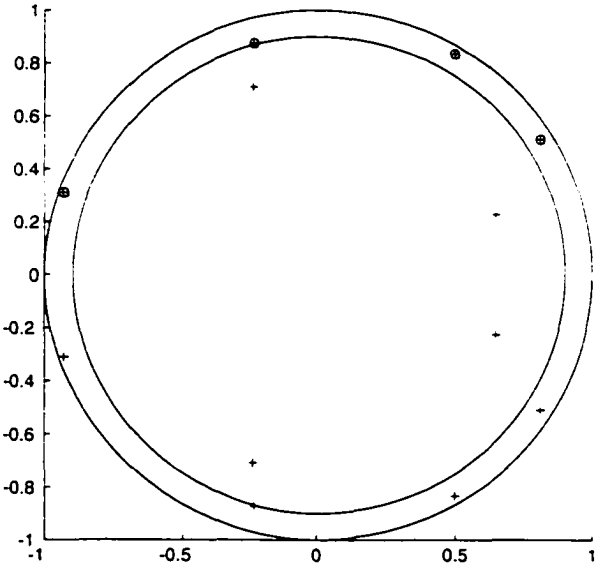


Figure 4.7: The poles of the voiced frame of a male speaker saying 'a'

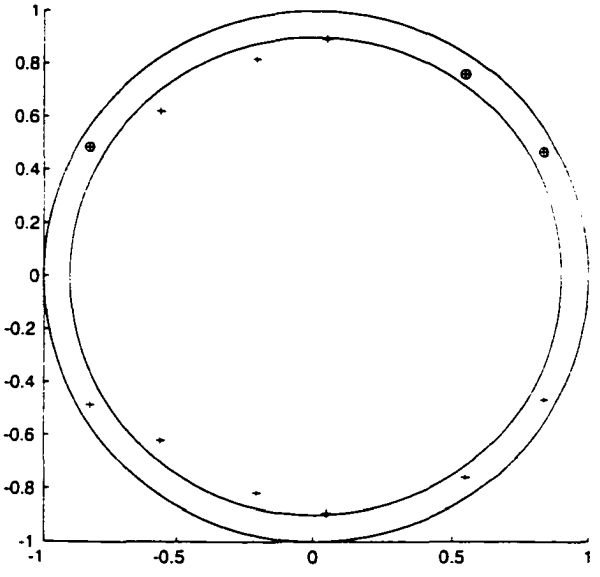


Figure 4.8: The poles of the voiced frame of a male speaker saying 'a' with -5 dB white gaussian noise added

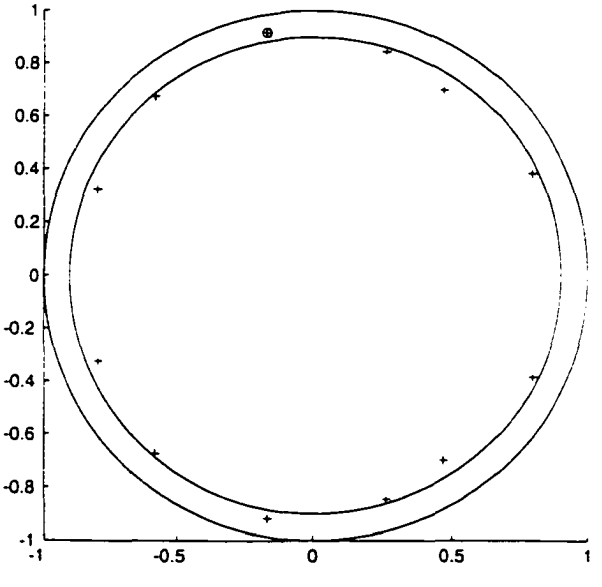


Figure 4.9: The poles of the unvoiced frame of a female speaker saying 'she'

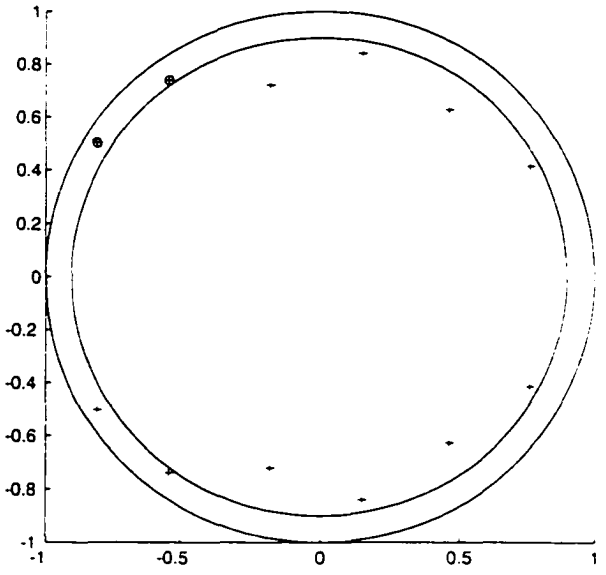


Figure 4.10: The poles of the unvoiced frame of a female speaker saying 'she' corrupted with -5 dB white gaussian noise

values. In [58], an adaptive smoothing approach which uses an iterative weighted averaging methods was implemented. This was achieved by repeatedly convolving the signal with a very small average mask weighted by a measure of the signal continuity at each point. The aim of these adaptive smoothing approaches is to average the samples near discontinuities with the point of discontinuity. The approaches utilizes a windowed filter such that two points of discontinuities are not averaged. The repeated averaging process forces discontinuities to belong to one of the windowed regions thereby enhancing these discontinuities. This principle has been adopted in the AGDF algorithm to smooth the speech residual in voiced frames in order to enhance the pitch peaks. In image processing, an adaptive algorithm is not used to

control the parameters of the GDF. This makes the AGDF novel in its approach to smooth a speech signal.

One of the major drawbacks with many adaptive approaches is that the convergence properties are not known *apriori*. In the analysis of the speech signal, it is well known that voiced signals can be considered quasi-stantionary [41]. In fact, based on [57], it is well known that within a short analysis windows, the pitch period will vary by up to $2\% - 10\%$ between two consecutive pitch periods. In noisy environments, we can expect this variation to be even greater since noise can actually displace the location of the pitch pulse. This statistical information within the analysis window can be used to control the spatial bandwidth of the GDF expressed in equation 4.3 and thus, the iteratively smooth the speech within the analysis window.

In order to optimize the spatial bandwidth of equation 4.3 to adaptively smooth the signal without destroying the pitch location information, it is necessary to tweak the parameters of the GDF. All three parameters, c_0, c_2 and σ, all affect the spatial bandwidth of the function. It can be noted that c_2 is always negative and c_0 is always positive to reflect the nature of the GDF which is the difference between the gaussian and its second derivative. In order to simplify the gaussian derivative function in equation 4.3, it is possible to rewrite the function by setting $c_0 + c_2 = 1$. The corresponding equation is

$$g_D(x/\sigma) = c_0(h_0(x/\sigma) + h_2(x/\sigma)) - h_2(x/\sigma) \tag{4.6}$$

By simplfying the equation, it is now less computationally intense to optimize the parameters of the GDF, since we have reduced the number of parameters to be optimized from three to two.

The spatial width of f(x) can be calculated using the following relationship:

$$(\Delta x)^2 = \frac{\int_{-\infty}^{\infty}(x-\mu_x)^2|f(x)|^2dx}{\int_{-\infty}^{\infty}|f(x)|^2dx} \tag{4.7}$$

Based on the property of the gaussian derivative function, μ_x, the mean of the function is zero. From equation 4.7, it is possible to express the spatial bandwidth of equation 4.6 as (See Appendix A for proof):

$$\Delta x = \tau^{1/2}\sigma \tag{4.8}$$

where

$$\tau = \frac{1+3\gamma+5\gamma^2}{2+2\gamma+1.5\gamma^2} \tag{4.9}$$

and

$$\gamma = \frac{c_0-1}{63c_0+1} \tag{4.10}$$

It is now possible to find the effects of the parameters c_0 and σ on the spatial bandwidth by performing a sensitivity analysis. The sensitivity of a parameter on a

function is described as the ratio of the percentage change of the function over the percentage change of the parameter. From [54], the following equations has been adapted for this purpose. For the spatial bandwidth expressed in equation 4.8, we can be express the following sensitivity expressions:

$$S_{\sigma}^{\Delta x} = \frac{\partial \Delta x}{\partial \sigma} \frac{\sigma}{\Delta x} \tag{4.11}$$

and

$$S_{c_0}^{\Delta x} = \frac{\partial \Delta x}{\partial c_0} \frac{c_0}{\Delta x} \tag{4.12}$$

Based on the sensitivity analysis discussed in detail in Appendix C, it is clear that varying the scaling function is directly related to changing the spatial bandwidth of the filter. The parameter, c_0, has a limited but larger effect at very low values of c_0 (below $c_0 = .1$). However, at larger values of c_0, there is very limited effect on the bandwidth (See Figure C.1 in Appendix C). Therefore, $|S_{c_0}^{\Delta x}| \ll |S_{\sigma}^{\Delta}|$ since $max|S_{c_0}^{\Delta x}| = .16$. This is significant since it is more important to optimize the spatial bandwidth by varying the scaling parameter σ. Varying the parameter c_0 does not affect the bandwidth considerably except at low values. Furthermore, $c_0 = .65$, gave the best results in noisy conditions for pitch detection.

The aim of this algorithm is to generate the best σ parameter of the GDF that would optimally smooth and thus enhance the pitch peaks based on statistical information

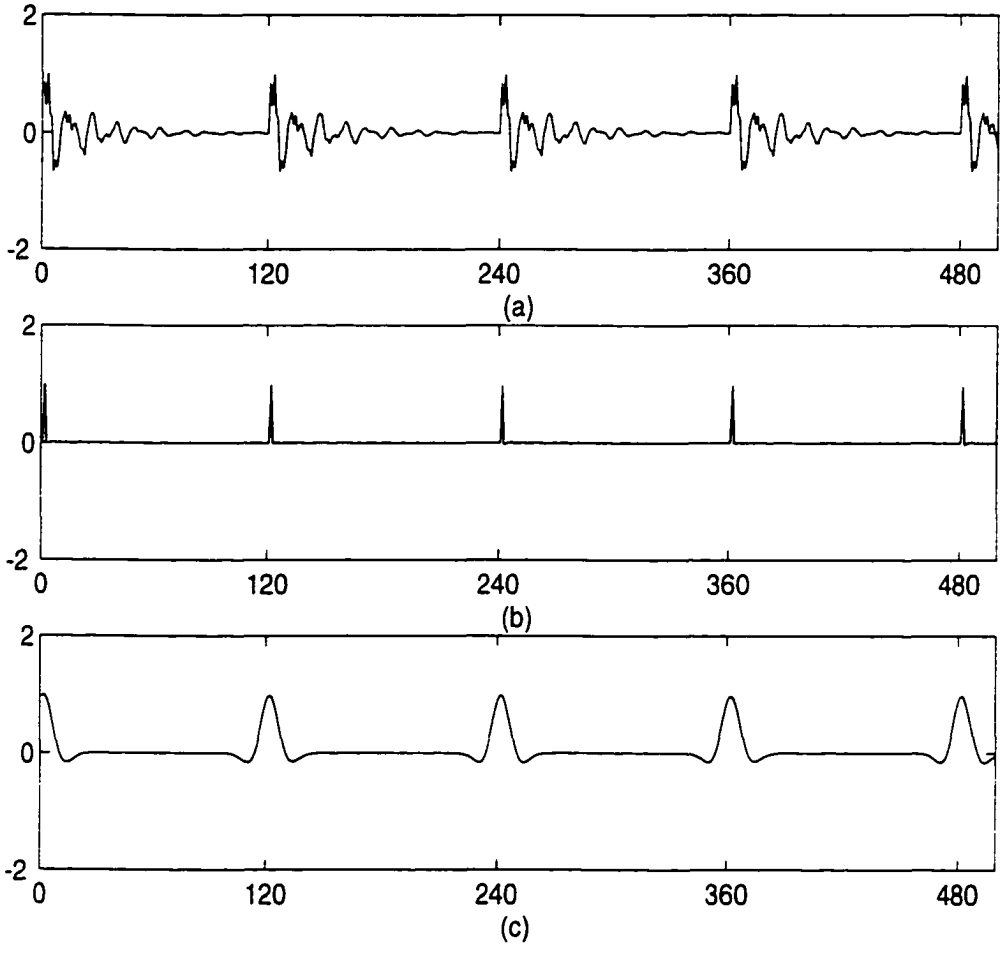


Figure 4.11: a) Synthesized signal /a/ b) LP residual of a) and c) Results of the AGD filter on b)

generated by taking initial estimates of the pitch information in a noisy signal. This is based on the principle that in the spatial domain, an GDF averages the samples within the spatial bandwidth of the filter. Therefore, the stopping criterion of the algorithm is to smooth just enough to make the standard deviation of the differences of the pitch peaks very low. A threshold of 20% of the average estimated pitch period, is based on the limit in variations experienced in a short speech window by a single speaker voiced frame as discussed earlier. Fig. 4.11 illustrates the affects on AGD filtered signal.

The AGDF algorithm for pitch determination can be summarized in the following steps:

1. The speech is windowed into 30 ms frames with an overlap of 20 ms.
2. The frame is passed through a silence/voice/unvoice detector described in the previous section. Only voiced frames are processed in the next step.
3. The voiced frame is LP filtered to remove the formant information.
4. For the GDF function $g_D(x/\sigma)$ (see Eq. (4.3)), set initial guess for the GDF scaling parameter at $\sigma = 3$ and $c_0 = .65$.
5. The GDF function is convolved with the LP residual frame.
6. Find the absolute maximum value of $r(n)$ and denote it as r_{max}. A set of estimated pitch periods of $r(n)$ are taken by (1) picking the local peaks of $r(n)$

for which $|r(n)| > r_{max}/2$, (2) finding the corresponding time indices of these local peaks and (3) finding the difference in the time indices between successive local peaks. For example, if two s uccessive peaks of $r(n)$ are at $n = 10$ and $n = 75$, the estimated pitch period is 65 samples.

7. Discard pitch period estimates below 3 ms and above 40 ms. There are two peaks that will have led to the discarded estimate. Of these two peaks, ignore the pitch peak that has a lower absolute signal amplitude. Recalculate the pitch period as in the previous step.

8. From the set of pitch period estimates, calculate an average pitch period and a standard deviation. Check whether the standard deviation is above 20% of the average pitch period. The standard deviation reflects how much the pitch period varies in a particular segment. For clean speech, we can expect a maximum variation of about 10% above and below the average pitch period. For noisy speech, the variation is much more and in this case, we have to repeatedly apply the AGDF with an updated σ. If the standard deviation is above 20% of the average pitch period, the GDF function is recomputed using the new value $\sigma = \sigma + i\Delta\sigma$ where $\Delta\sigma$ is a small number (usually around 0.1) and we go back to Step 3. The number of iterations, i, is set to 20 in this algorithm. This repetitive application of the GDF is equivalent to an adaptive way of setting σ and hence, the acronym AGDF. If the standard deviation is below 20% of the average pitch period, the pitch periods are approximately

equally spaced indicating that the pitch period estimation is good and the algorithm terminates.

9. If the number of repetitions exceed 20 then the pitch information at then the 20th iteration will be the final pitch estimate.

10. Check for doubling and tripling. For a segment of voiced frames, check basic statistics on the pitch estimates $\hat{p}$. If a pitch estimated $\hat{p}_i$ is between $n*(avg.\hat{p} - .2 * stdv.\hat{p})$ and $n * (avg.\hat{p} + .2 * stdv.\hat{p})$ where $n = 2, 3$ then reestimate new pitch period by taking $\frac{\hat{p}_i}{n}$.

Although both the DyWT and AGDF algorithms attempt to smooth the signal, there is a fundamental difference which needs to reiterated. The AGDF algorithm does not use the multiresolution property to smooth the signal or cancel the noise. Instead, it relies on the spatial bandwidth parameter to be set adaptively to control the level of smoothing based on the statistical properties within the voiced signal segment. The basic AGDF algorithm for pitch estimation is shown in Fig. 4.12.

4.3 Results and Discussion

We tested the accuracy and robustness of the AGDF algorithm against the DyWT algorithm for pitch detection. In addition, we demonstrate the use of the AGDF algorithm on continuous conversational speech. In order to test the robustness of

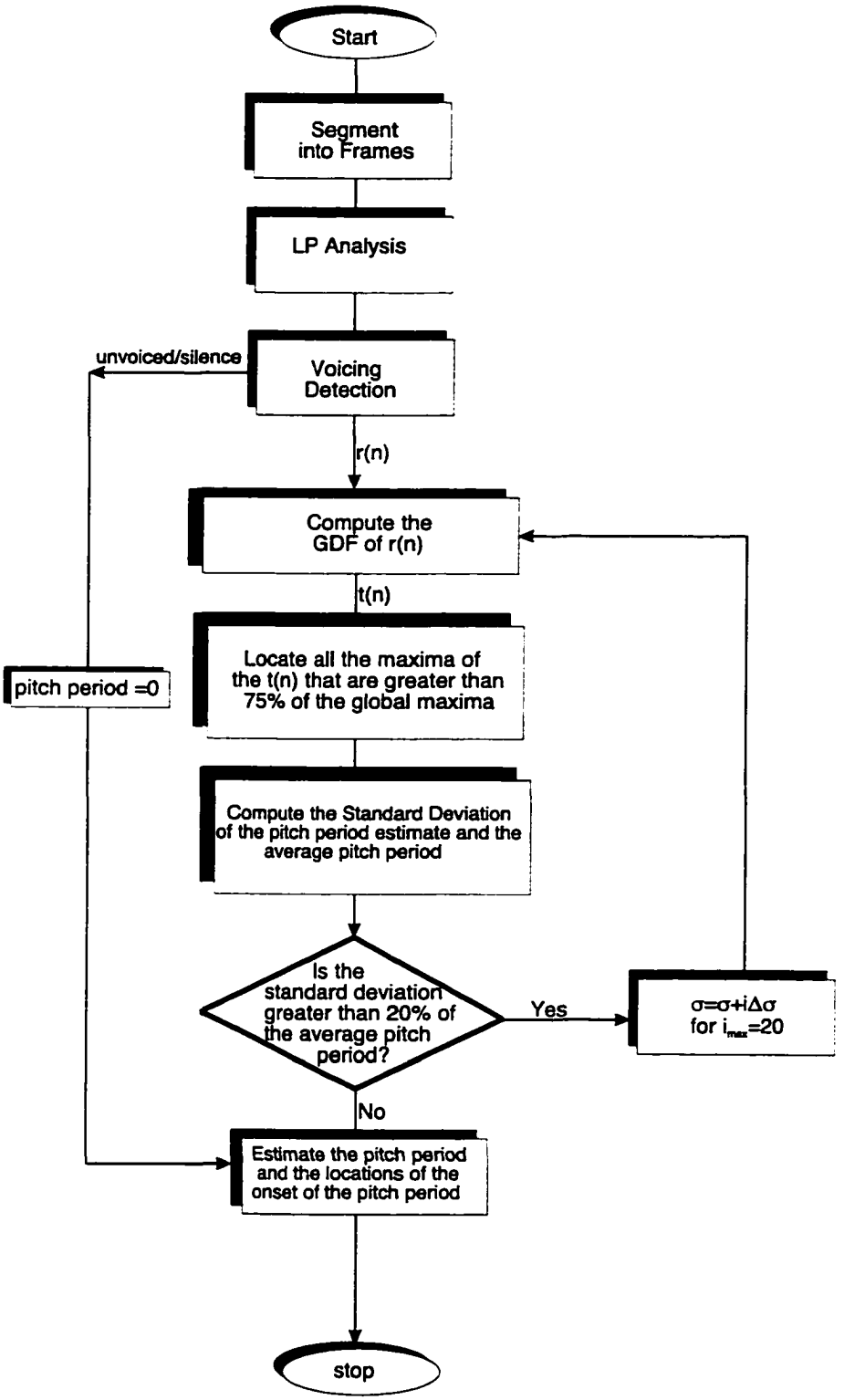


Figure 4.12: The AGDF algorithm for pitch estimation

the algorithm, we have synthesized voiced 8 kHz sampled signals as was done in [57][55]. This is convenient since synthesized speech allows for the pitch period to be set *apriori* and also simulates the sharp discontinuities encountered in glottal pulses. Unfortunately, there is no standard criterion for quantifying the performance of pitch determination algorithms. Hence, we use our own measure of relative accuracy as described later. This allows us to test the accuracy of our new algorithm against the DyWT method. To test the robustness of the algorithms in noisy conditions, the synthetic speech signal was corrupted with three different types of noise: white gaussian, 'colored ' and 'babble'. The colored noise was generated by passing white gaussian noise through a recursive linear predictive filter computed from a frame of speech corresponding to a sustained vowel. Babble noise was generated by combining the speech signals (different utterances) of 10 interfering speakers (similar to the noise heard at a cocktail party).

In our experiments for AGDF pitch detection we have experimented with several frame sizes. In event based pitch detection with the AGDF, however, the frame sizes are not very significant since we are locating the instant of glottal closure. This is a clear advantage over many pitch detection methods such as the cepstrum and the autocorrelation methods that require short frame sizes in order to estimate the average pitch. The basic requirements for the AGDF is that there should be at least two pitch peaks in a chosen segment. In our experiments, we have found that frame sizes of 30 ms with a frame overlap of 20 ms gave the best results. Partially

overlapping of the consecutive frames are importantfor enhanced performance of the AGDF algorithm.

In Fig. 4.13. the results for both algorithms are illustrated for different conditions of noise and varying signal-to-noise ratios (SNR) (from $-10\ dB$ to $30\ dB$). The relative accuracy (*rel. acc.*) was determined by the following relationship:

$$rel.\ acc. = \left(1 - \frac{1}{N}.\sum_{k}^{N} abs(\frac{x - y_k}{x})\right) \times 100\% \tag{4.13}$$

Where x is the true pitch period of the synthesized speech. y_k is the estimated pitch periods and k is the number of estimated pitch periods in a synthesized speech segment.

In the first study. the aim is to determine the relative accuracy and robustness of the DyWT and AGDF algorithms under varying conditions of white gaussian. colored and babble noise. We use a synthesized speech signal of the vowel /u/ with a pitch period of 25 ms. Clearly. the AGDF algorithm performs better than the DyWT pitch detector for white gaussian noise and babble noise at low SNRs. For high SNRs. both algorithms show comparable results. For white gaussian noise. the performance of the DyWT method starts to drop steeply at an SNR of $-5\ dB$. For babble noise. the performance of the DyWT method starts to drop steeply at an SNR of 0 dB. For colored noise. the two algorithms are comparable for all SNRs. However. the AGDF still performs slightly better.

In a second experiment to determine robustness of the algorithms for high pitch synthesized speech, a synthesized signal /o/ with a pitch period of 10 *ms* was tested and the results are illustrated in Figure 4.14. In the case of white gaussian noise, the rel.accuracy of both the DyWT algorithm begins to decrease at about 5 *dB*. The AGDF algorithm outperforms the DyWT algorithm for pitch detection until about -10 *dB* when both algorithms gives near equal performance. In the case of babble and colored noise, the AGDF still outperforms the DyWT algorithm at up to -5 *dB* snr. However, at -10 *dB* the DyWT slightly outperforms the AGDF algorithm. Comparing the performance of low and high pitch speakers, both the AGDF and DyWT algorithms perform much better for low-pitch speakers than for the high-pitch speakers. The reason appears to be the fact that both algorithms use smoothing. In the process of smoothing, controlling the smoothing of closely position peaks in the presence of noise is very difficult to accomplish. In the AGDF algorithm, two peaks closely positioned together may averaged together or might be diminished. When applying the peak algorithm the pitch peaks may not be selected.

In terms of real speech, we first demonstrate the ability of the AGDF to classify the vowel 'a' spoken by a male speaker continuously. Since this is real speech, the pitch period of the speaker is not known *apriori* and can vary slightly within an analysis window. The AGDF was tested, firstly with no noise added to the signal and then with 0 *dB* and -5 *dB* white gaussian noise added. The results are shown in Fig. 4.15 and Fig. 4.16. The results clearly indicate that the AGDF maintains a consistent

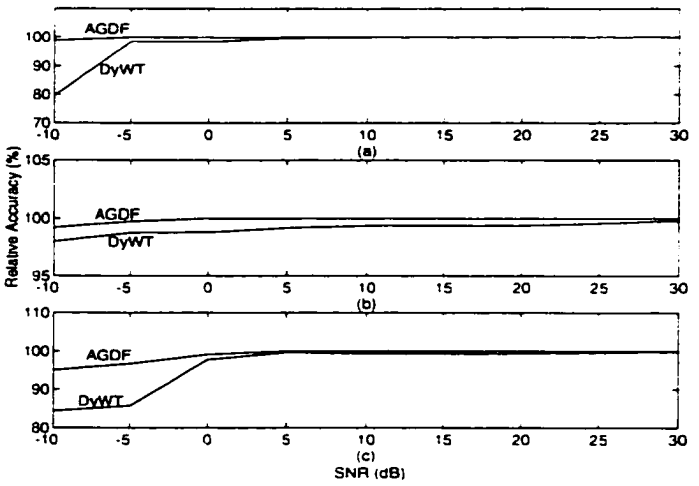


Figure 4.13: SNR vs. Relative Accuracy of AGDF and DyWT for synthesized signal /u/ with a pitch period of 25 *ms* a) White Gaussian noise b) Colored noise c) Babble noise

pitch track in the ideal case of no noise added. However, in the presence of -5 dB white gaussian noise, the AGDF varies slightly from its ideal conditions.

In the final demonstration, we provide an example of the use of the AGDF algorithm for real conversational speech. We consider the speech signal *"greasy wash water"*, spoken by a female American speaker from the TIMIT database which was sampled at 8 kHz. The signal itself consist of several periods of silent, unvoiced and voiced segments. The algorithm will determine the pitch period for all voiced segments and set the pitch period to zero for unvoiced and silent segments. The window sized used in this example is 30 ms with an overlap of 20 ms. The overlap is crucial in obtaining a correct pitch track especially when there are voiced-unvoiced transitions

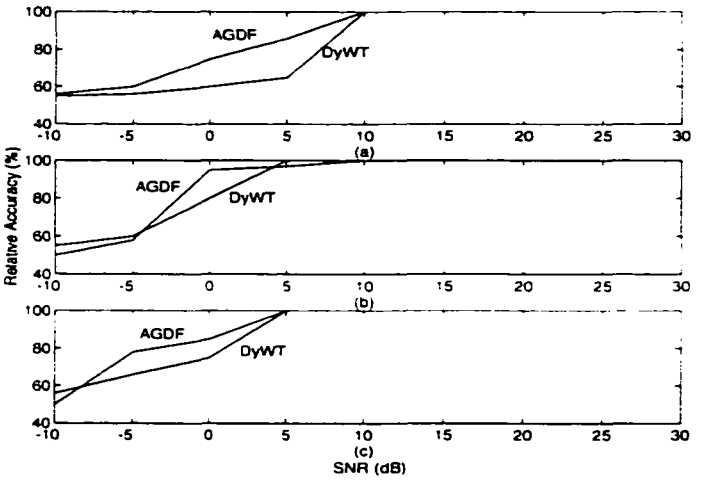


Figure 4.14: Relative Accuracy vs. pitch period of AGDF and DyWT for synthesized signal /o/ with a pitch period of 10 ms a) White Gaussian noise b) Colored noise c) Babble noise

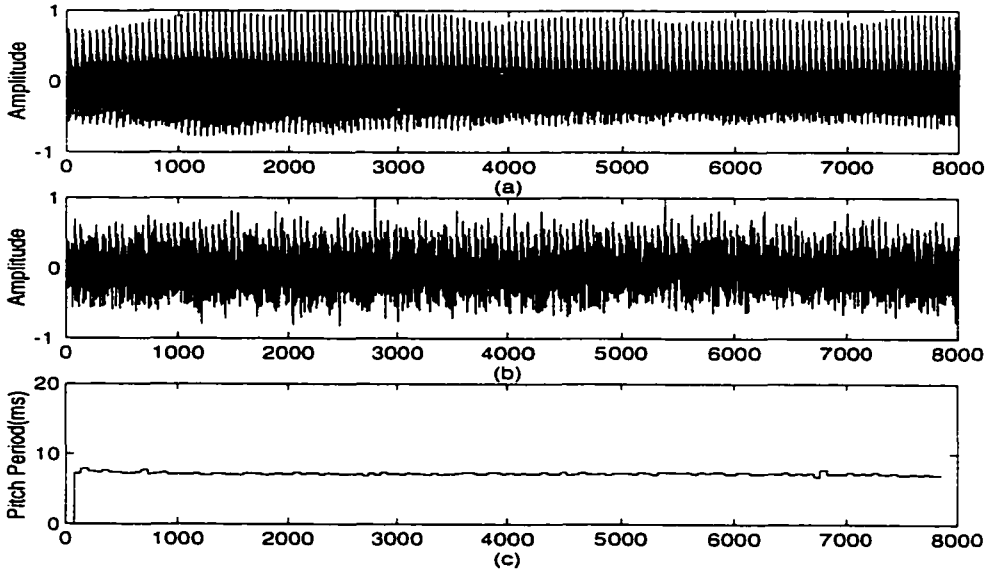


Figure 4.15: a) Male speaker uttering 'a' with $0dB$ white gaussian noise added in and b) the pitchtrack of a)

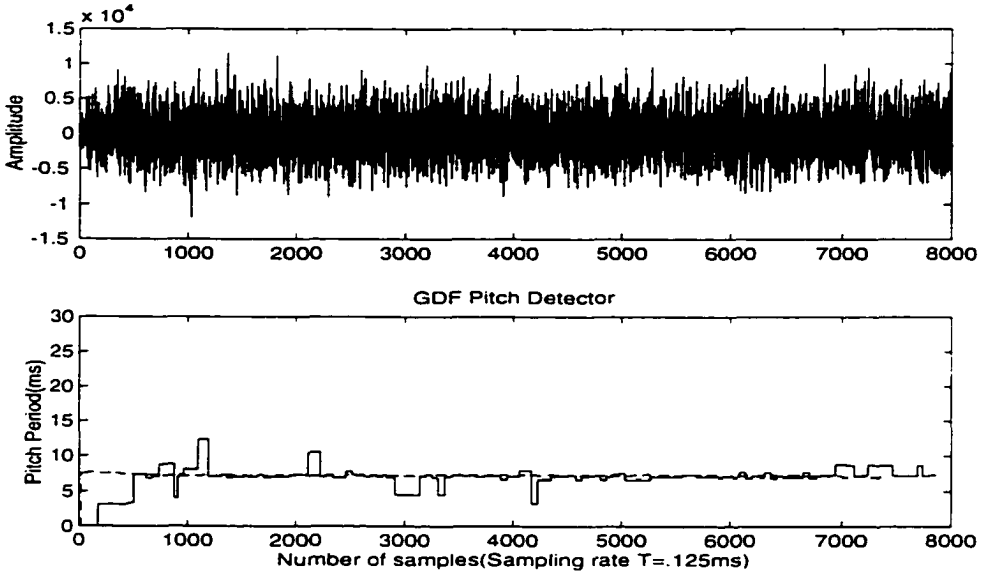


Figure 4.16: a) Male speaker uttering 'a' with $-5dB$ white gaussian noise added in and b) the pitchtrack of a)

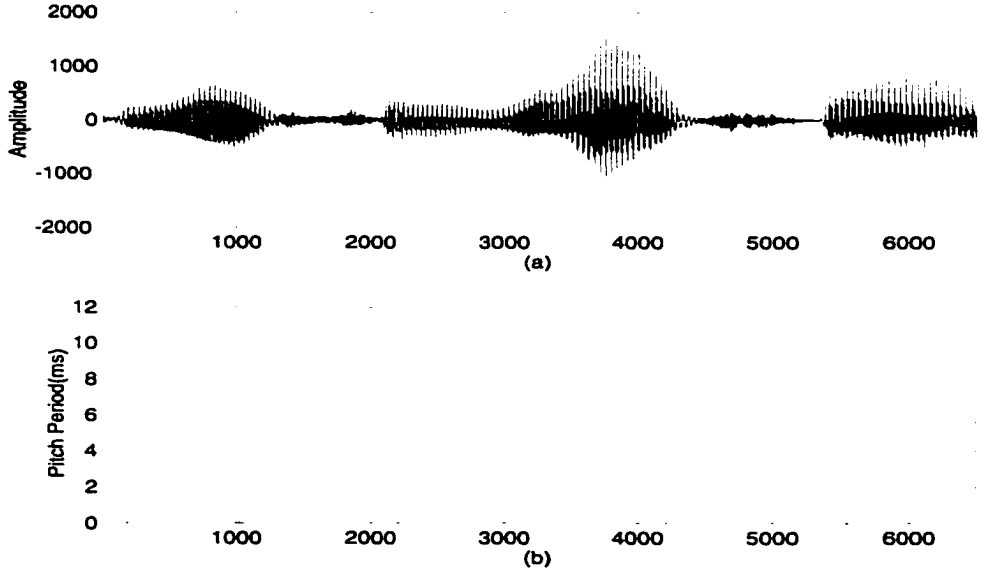


Figure 4.17: a)'greasy wash water' spoken by a female speaker and b) the pitchtrack of a)

or vice versa. The pitch track of the AGDF for 8 kHz real conversational speech is illustrated in Fig. 4.17

4.4 Conclusion

In this chapter, an adaptive gaussian derivative filter (AGDF) has been introduced for the determination of the pitch period in speech. The quadratic spline DyWT Wavelet was chosen as a benchmark for comparison since it was shown by Kadambe, et al [55] to outperform all other pitch detection algorithms for low and high pitch speakers in noise. When the AGDF detector was compared with the DyWT pitch detector, it is shown that the AGDF is more robust than the DyWT pitch detector for white noise and babble noise conditions at low SNRs. This results also illustrates the AGDF algorithm can be used for the accurate determination the pitch period in conversational speech. In chapter 6, future work is proposed for the AGDF algorithm to further enhance its performance.

Chapter 5

New Adaptive Comb Filtering Methods For Speech Enhancement

5.1 Introduction

The objective of any speech enhancement system is to either improve the signal-to-noise ratio, to increase speech intelligibility or reduce listener fatigue, etc. In noisy environments, such as offices, interfering background noise poses many problems for speech processing applications such as speech coders, speaker and speech recognition systems, etc. The performance of speaker and speech recognition systems, in particular, are badly degraded in remote applications where the signal risks being

contaminated with noise. These systems are usually developed under laboratory conditions where the environment does not represent real-world situations which are variant and cannot be predicted. The problem is known as the "mis-match" problem and has been encountered earlier in Chapter 3 for the case of cochannel interference. In this problem, differences occur in the training and testing phases of the algorithm which prevents the application from being robust.

The spectral subtraction method is the most direct method used for enhancing noise [17]. This method is based on the statistical assumption that the power spectral density function of the signal that is contaminated with uncorrelated noise is equal to the power spectral density of the noisy signal plus the power of the noisy process. Therefore, if $|\bar{R(\omega)}|^2$ is the power spectral estimator of the noisy process, $|\bar{Y_i(\omega)}|^2$ the power spectral estimator of the input signal for the $i-th$ analysis frame, and $|\hat{X_i(\omega)}|^2$ is the power spectral estimator for the enhanced signal for the $i-th$ analysis frame, then the spectral subtraction process can be expressed as

$$|\bar{X_i(\omega)}|^2 = \begin{cases} |\bar{Y_i(\omega)}|^2 - \alpha \cdot |\bar{R_i(\omega)}|^2, & if |\bar{Y_i(\omega)}|^2 - |\bar{R_i(\omega)}|^2 > \beta \cdot |\bar{R_i(\omega)}|^2 \\ \beta \cdot |\bar{R_i(\omega)}|^2, & \text{otherwise} \end{cases}$$

where $\alpha > 1$ and $0 < \beta \ll 1$ are constants that minimize the spectral spikes that are known as musical noise. This speech enhancement technique is used as a preprocessor to many speech processing applications that are degraded by noise. The level of performance of these applications have been shown to improve dramatically after being enhanced by this technique.

The problem with this spectral subtraction technique however, is that background noise is estimated during periods of silence in the utterance. This assumes that the utterance of the speaker is long enough to study the long-term characteristics of the signal. In many applications such as speaker recognition systems the utterances can be very short, therefore, estimation of background noise may not be very accurate. In many cases, the different types of noises and SNRs make the estimation of background noise in speech very difficult.

In order to avoid the afore mention problems, speech enhancement of the pitch information in the voiced segments of speech have been suggested by [5][7]. This principle is based on the fact that voiced sounds are periodic in nature. Specifically, the periodicity of a time waveform manifest itself in the frequency domain as harmonics with the fundamental frequency corresponding to the period of the time waveform. Also, most of the energy of a periodic signal is concentrated in bands of frequencies. Since the interfering signals has noise over the entire frequency bands, to the extent that accurate information about the fundamental frequency or pitch is available, a comb filter can reduce the noise while preserving the signal.

In Chapter 4, it was previously discussed that the pitch information present in the voiced segment of speech varies up to 10% from period to period. Therefore, in order to accurately preserve the pitch information, it is necessary to apply an adaptive comb filtering method. In [5], an Adaptive Comb Filter (ACF) gave an improved

signal when known pitch information was used to enhance nonsense sentences test material that was degraded by wide-band random noise. The results of the test showed that even with accurate pitch information, the adaptive filtering technique tends to decrease intelligibility at various S/N ratio. However, despite the decreased intelligibility, speech processed by an adaptive comb filter sounds "less noisy" due to the capability of the algorithm to increase the S/N ratio. In [38], it was found that adaptive comb filtering of voice frames gave improved performance of speaker ID systems. Therefore, it can be expected that with the use of accurate and robust pitch detection results of the DyWT or AGDF pitch detector, noisy speech will be enhanced.

5.2 Theory

In this section, we discuss the basic theory of Adaptive Comb Filtering (ACF) and novel approaches of integrating both the DyWT and AGDF pitch detection algorithms.

1 Adaptive Comb Filtering

The basic aim of adaptive comb filtering is noise reduction without much speech distortions. The operation of adaptive comb filtering can be explained by considering

its unit sample response over one pitch period [7]:

$$h(n) = \sum_{k=-L}^{L} a_k \cdot \delta(n - N_k) \tag{5.1}$$

where $h(n)$ is the unit sample response, $\delta(n)$ is a unit sample function, the length of the filter is $2L+1$ pitch periods, a_k is the filter coefficient that satisfies $\sum_{k=-L}^{L} a_k = 1$, and N_k is given by the following equations:

$$N_k = \begin{cases} -\sum_{l=k}^{-1} T_l, & \text{for } k < 0 \\ 0, & \text{for } k = 0 \\ \sum_{l=0}^{k-1} T_l, & \text{for } k > 0 \end{cases} \tag{5.2}$$

T_k corresponds to the particular pitch period which contains the point of speech waveform that is multiplied by the filter coefficients a_k which are derived by Frazier in [5]. The filter coefficients remain unchanged, however, the N_k is usually updated every pitch period based on the pitch information of the speech waveform processed. Therefore, the algorithm will benefit from an accurate pitch detection algorithm so that the information of the impulse response of the filter can be accurately determined. The filter is applied over a $2L+1$ pitch periods to the extent that the speech waveform is periodic. It is expected that the speech samples will add constructively, but the noise samples will sum toward zero.

In terms of the filter coefficients which determines the unit sample response of the ACF, a_k was chosen based on qualitative results of Frazier, et. al. in [5]. Several

experiments were carried out to obtain the best window function based on informal listening tests. Frazier considered four different kinds of unit sample response shapes corresponding to rectangular, Hamming, Hanning and Blackman windows. It was reported that the rectangular windows gave the worse performance in terms of distortions whereas, all other methods gave more or less the similar results. In the ACF algorithm developed in this thesis, the Hamming window was chosen. The Hamming window shape can be described by the following equation:

$$a_k = \frac{.54 + .46\cos(2\pi k/(2L+1))}{\sum_{k=-L}^{k=L} .54 + .46\cos(2\pi k/(2L+1))} \qquad -L \leq k \leq L \tag{5.3}$$

In order to prevent what is considered the "overload problem" which occurs when T_0 is longer than any of the $T_i (i \neq 0)$, the filter is turned off. In that instance, one pitch period is covered by some filter coefficient a_k while a_0 is applied to the full length of T_0. For such cases, the value of a_k was made to be zeros for the portion that exceeds one local pitch period. The filter is also turned off during periods of voiced/unvoiced transitions.

Mathematically, it can be shown that though the ACF method has been shown to decrease intelligibility, applications which rely on increased S/N ratio e.g. Speaker ID systems will benefit tremendously due to the capability of its noise reduction abilities. For voiced sounds, the approximate S/N ratio increase due to the ACF can

be derived in the following manner. The output of the ACF can be represented by

$$y(n) = \sum_{k=-L}^{L} a_k \dot{x}(n - N_k) \tag{5.4}$$

where $x(n)$ is degraded speech, $y(n)$ is processed speech and N_k is the point of filter coefficient a_k. Since $x(n) = s(n) + w(n)$ where $s(n)$ is speech and $w(n)$ is the degrading source, we get

$$y(n) = \sum_{k=-L}^{L} a_k \dot{s}(n - N_k) + \sum_{k=-L}^{L} a_k \dot{w}(n - N_k) \tag{5.5}$$

Assuming that $s(n) = \sum_{k=-L}^{L} a_k \dot{s}(n - N_k)$, which is the basis of comb filtering.

$$y(n) = s(n) + \sum_{k=-L}^{L} a_k \dot{w}(n - N_k) \tag{5.6}$$

From equation 5.6,

$$S/N_{y(n)}(dB) = 10 \log \frac{\sum_n s^2(n)}{E[\sum_n (\sum_{k=-L}^{L} a_k \dot{w}(n - N_k))^2]} \tag{5.7}$$

$$= 10 \log \frac{\sum_n s^2(n)}{E[\sum_n (\sum_{k=-L}^{L} a_k \dot{N}_0)]} \tag{5.8}$$

where $N_0 = E[w^2(n)]$

Since $x(n) = s(n) + w(n)$,

$$S/N_{x(n)}(dB) = 10 \log \frac{\sum_n s^2(n)}{E[\sum_n N_0} \tag{5.9}$$

From equation 5.8 and equation 5.9,

$$\begin{aligned} S/N_{Increase} &= S/N_{y(n)} - S/N_{x(n)} \\ &= -10\log\sum_{k=-L}^{L} a_k^2 \end{aligned} \tag{5.10}$$

Therefore, the approximate S/N increase is due to the filter lengths determined by L of the ACF. A filter length of 7 and 13 pitch periods will give a S/N increase of approximately $7dB$ and $10dB$, respectively. Therefore, it is expected that the ACF will reduce the noise in a signal.

2 Algorithms

In the first step of the ACF algorithm, the speech is segmented into short frames of 240 samples and a 160 samples overlap for $8kHz$ sampled speech. The short frames are necessary in order to use the voiced/unvoiced/silence frame selection algorithm described in Chapter 4. The silent frames and unvoiced frames are treated similarly and have been attenuated by multiplying the frames by a factor of .4 as was done in [7]. Two voiced frames of interest are grouped to form a 60ms segment and then processed to determine the pitch information. This grouping of frames is to ensure that the ACF algorithms will be given enough information to construct the filter. In this pitch information determination, two new robust pitch detection have been

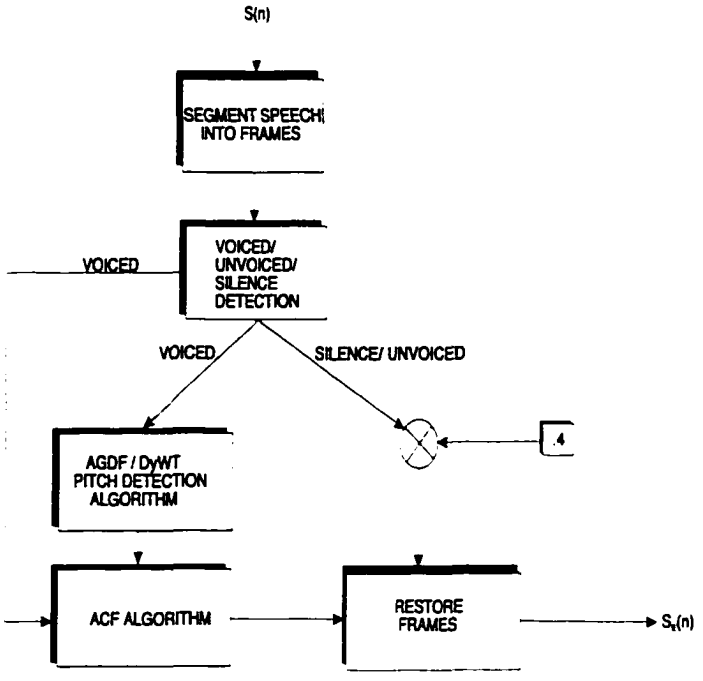


Figure 5.1: The Adaptive Comb Filtering Algorithm Using The AGDF or the DyWT Pitch Detection Methods. The input speech is denoted by s(n) while the enhanced speech is denoted as $S_E(n)$

introduced for the first time, to the author's knowledge, for speech enhancement. The AGDF robust pitch detection or the DyWT pitch detector can be used (See Chapter 4). An option of using the fused results of both algorithms is also possible. The objective is to obtain the most accurate pitch information that can be obtained from a noisy signal, so that this information can be fed to the ACF algorithm described in the previous section. In a final step, the filtered frames are restored in an ordered fashion to synthesize the enhanced signal.

In the next few examples, the performance of ACF algorithm with the AGDF and DyWT pitch detection method is illustrated. In Fig. 5.2, a synthesized signal /a/ is corrupted with $0dB$ white ggaussian noise. The figure shows a clean signal in Fig. 5.2(a), the corrupted signal in Fig. 5.2(b) and the enhanced signal is shown in Fig 5.2(c). The results clearly demonstrates the ability of the ACF algorithm to enhance a corrupted signal. In a second example, 8 kHz speech of a female speaker saying 'e' taken from the TIMIT database is corrupted with 15 dB white gaussian noise. Figure 5.3 shows that the ACF algorithm in conjunction with the AGDF algorithm does a very good job in reducing noise and enhancing the pitch information. In the final example, the ACF algorithm used in conjunction with the DyWT is demonstrated for the first time for speech enhancement. A male speaker saying 'a' is corrupted with 10 dB colored noise (See Chapter 4). Figure 5.4 shows that the ACF used in conjunction with the DyWT pitch detection scheme is very effective in reducing noise and maintaining the integrity of the pitch information of

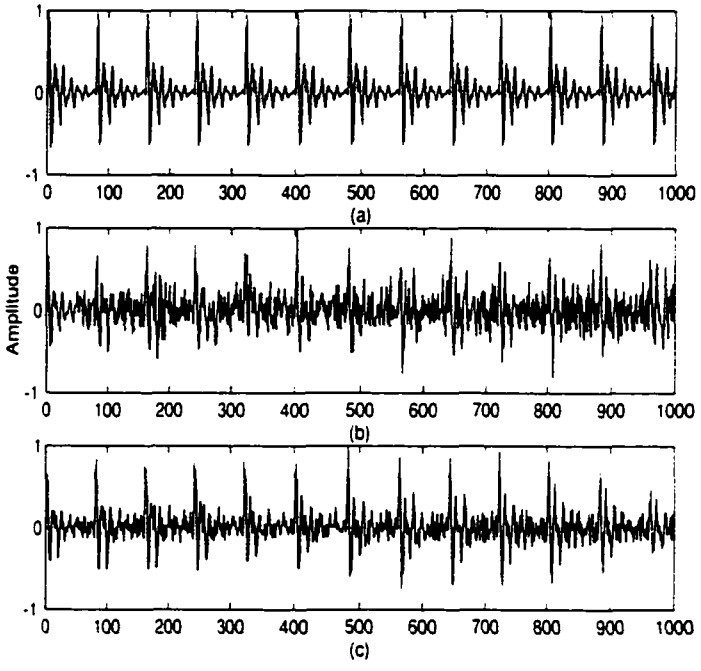


Figure 5.2: a) Synthesized signal /a/ b) The signal in a) corrupted with 0dB white gaussian noise and c) The ACF-AGDF enhanced signal

the signal in the presence of colored noise.

In the following section, the ACF is tested to qualitatively determine its ability under various conditions of noise.

5.3 Results and Discussion

In order to test the effectiveness of the ACF algorithms, it is necessary to use an objective measure. By applying parseval's theorem, it can be shown that for a pair of

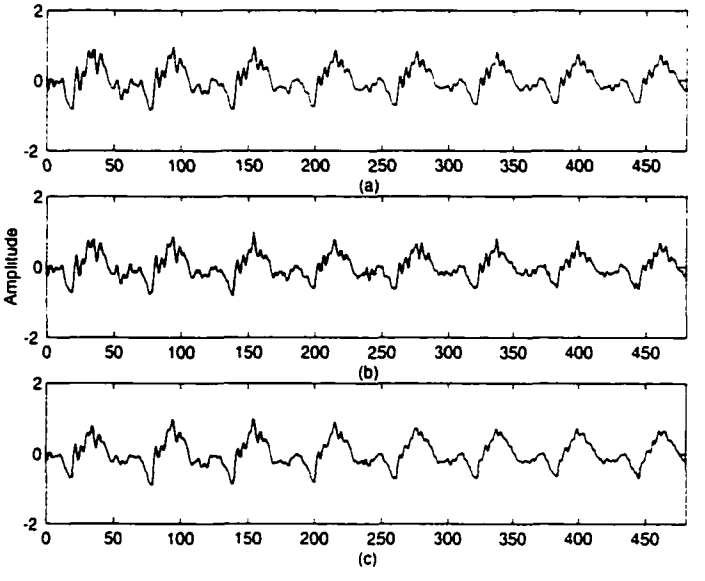


Figure 5.3: a) A female speaker saying 'e' b) The signal in a) corrupted with 15dB white gaussian noise and c) The ACF-AGDF enhanced signal

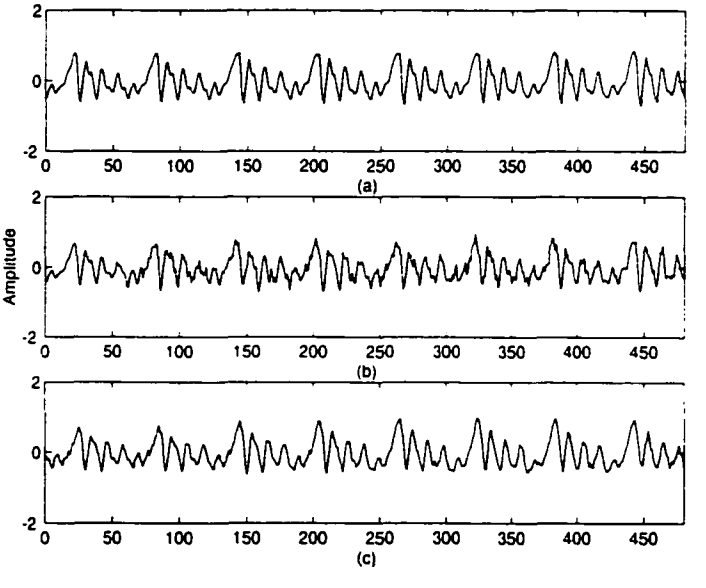


Figure 5.4: a) A male speaker saying 'a' b) The signal in a) corrupted with 10 dB colored noise and c) The ACF-DyWT enhanced signal

spectra $S(\omega)$ and $S'(\omega)$ that the L_2 cepstral distance of the spectra can be related to the rms log spectral distance. This measure is particularly important for Speaker ID systems since the cepstral feature is the most popular of all the speaker ID features. Furthermore, if a noisy signal can be enhanced such that the cepstral distance is lower or the spectra is closer to the original signal than the noisy signal then it can be concluded that a speaker ID will give better performance.

The cepstral distance (d_{cep}) measure is computed using the LP cepstrum analysis discussed in detail in Chapter 2. The measure can be defined as

$$d_{cep} \cong 2[\sum_{i=1}^{K}[C_x(i) - C_y(i)]^2]^{1/2} \tag{5.11}$$

where the C_x and C_y corresponds to the cepstrum of the input and output speech respectively.

The speech is corrupted with white gaussian, colored and babble noise at different SNRs. The different types of noise are discussed in more detail in Chapter 4. Also, the exact conditions are used for the AGDF and DyWT pitch detection as described in Chapter 4.

In the first experiment, the AGDF and DyWT pitch detection methods are tested in conjunction with the ACF method for speech enhancement of a synthesized signal corrupted by various levels of noise. In Figure 5.5-Figure 5.7, the d_{cep} results of the ACF-AGDF and ACF-DyWT are illustrated.

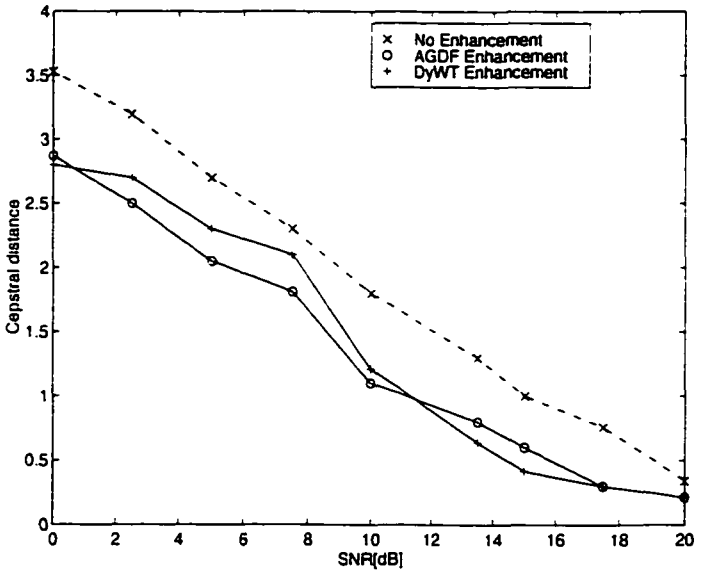


Figure 5.5: Cepstral distance measure of a synthesized signal /a/ corrupted with white gaussian noise

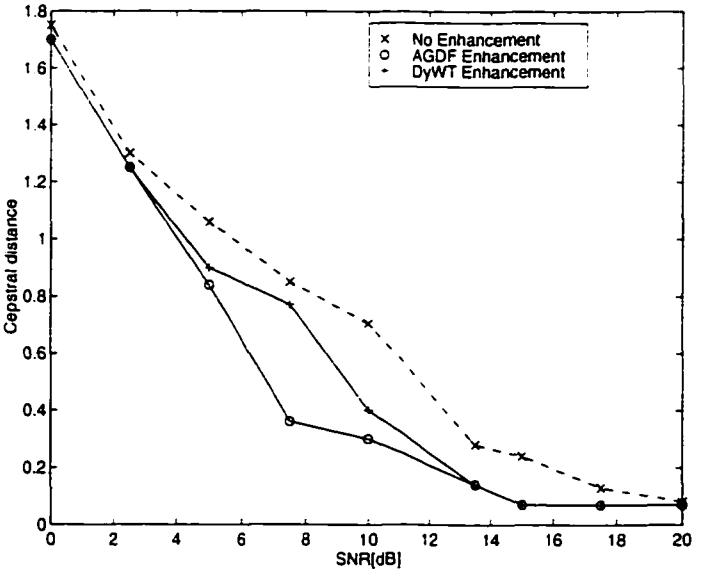


Figure 5.6: Cepstral distance measure of a synthesized signal /a/ corrupted with colored noise

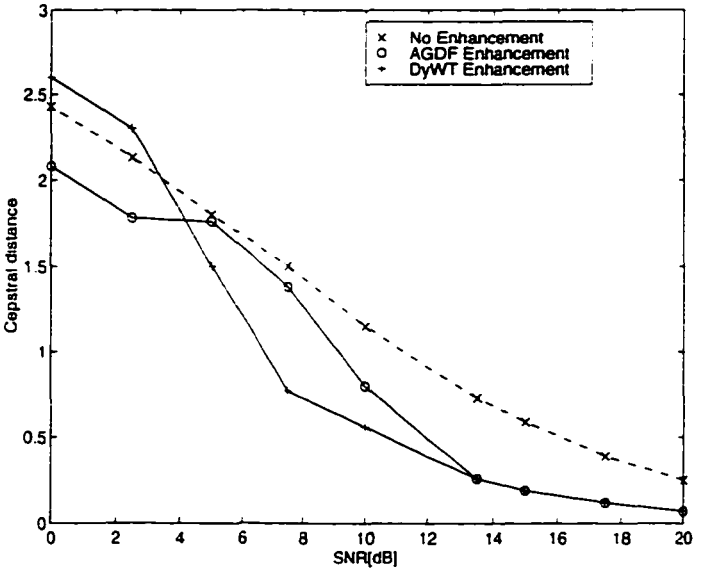


Figure 5.7: Cepstral distance measure of a synthesized signal /a/ corrupted with babble noise

The results show that in the case of a synthesized signal /a/ in the presence of white gaussian noise, the speech enhancement algorithms gave the most dramatic decrease in the cepstral distance. When babble and colored noise were added, there was a slight improvement due to the enhancement. In addition, the AGDF-ACF algorithm performs much better at lower signal to noise ratios than the DyWT-ACF algorithm. This confirms the fact established in Chapter 4 that at lower SNRs, the AGDF outperforms the DyWT pitch detection algorithm. The ACF algorithm relies on accurate pitch information to improve the signal to noise ratio.

In the second set of results, the effects of of ACF speech enhancement algorithms are examined related to the real speech signal of a female speaker saying 'e' extracted for

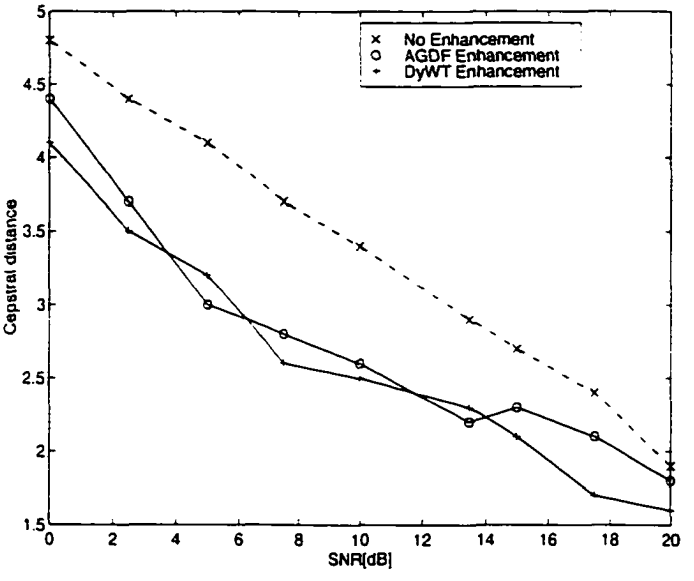


Figure 5.8: Cepstral distance measure of a female speaker saying 'e' corrupted with white gaussian noise

the TIMIT database. The signal is corrupted with white, colored and babble noise as shown in Figure 5.8-Figure 5.10 for varying noise levels.

Similarly to the synthesized speech, there was a dramatic lowering of the cepstral distance for the signal when corrupted with white gaussian noise. The results of the signal corrupted by babble and colored noise also gave very similar results to the synthesized /a/ which was previously discussed.

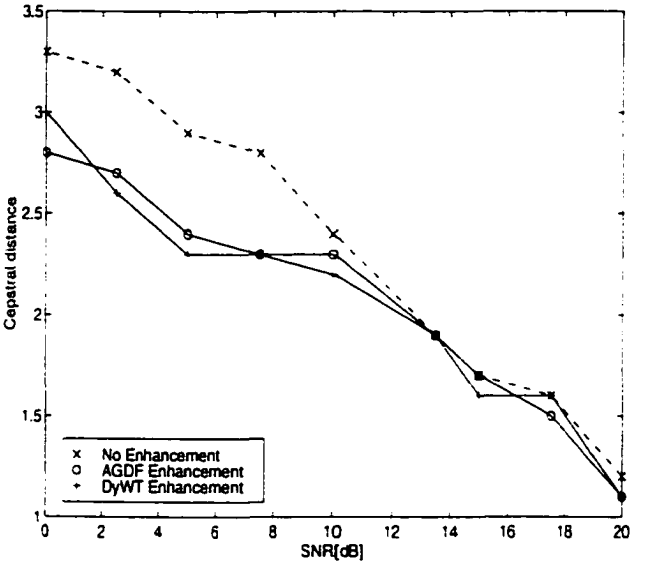


Figure 5.9: Cepstral distance measure of a female speaker saying 'e' corrupted with colored noise

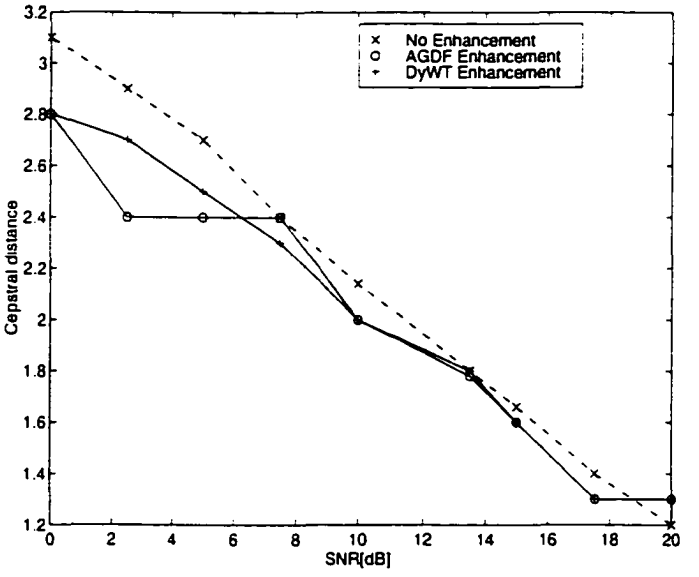


Figure 5.10: Cepstral distance measure of a female speaker saying 'e' corrupted with babble noise

5.4 Conclusion

The ACF algorithm used in conjunction with the AGDF and DyWT pitch detection algorithms have been demonstrated for the first time. As expected, the both algorithms showed great potential in reducing the distortions due to noise in a signal. The algorithms showed extremely good performance in the presence of white gausian noise. This confirms that the both AGDF and DyWT pitch detection algorithms performed very well on speech corrupted by white gaussian noise discussed in detail in Chapter 4. In addition, the both algorithms reduced the presence of noise in signals corrupted by babble and colored noise. These results are particularly important for Speaker ID systems which rely heavily on robust cepstral features for improved results.

Chapter 6

Conclusions

6.1 Summary Of The Results

In this thesis, three main topics have been addressed:

1. The problem of identifying temporal regions or frames as being either one-speaker or two-speaker speech was examined. This identification is important in making automatic speaker and speech recognition systems more robust and is based on feature extraction and subsequent classification as is done in pattern recognition. The research looked into both the closed set problem where the identity of the two interfering speakers are known *apriori* and the more difficult open set problem where the identities are not known (speaker independent).

For the feature extraction step, a new pitch prediction feature (PPF) was developed. This new PPF was compared with the Linear Predictive Cepstral Coefficients (LPCC) and the Mel Frequency Cepstral Coefficients (MFCC). The features were computed and classified on a frame by frame basis. We compared the performance of two classifiers, namely, the neural tree network (NTN) and vector quantizer (VQ). The results showed that in both the closed and open set cases, (1) the VQ was the better classifier and (2) the PPF outperforms both the MFCC and LPCC features. The superiority of the PPF came with the added benefits of using a scalar feature as opposed to the 12 dimensional vectorial LPCC and MFCC features and a lower VQ codebook size.

2. A new pitch detection algorithm based on an iterative adaptive smoothing approach using a derivative (GD) filter which is the sum of a zeroth and second order function was developed. The algorithms works under varying noise conditions, with variable pitch periods and for different speakers. The performance of the Dyadic Wavelet Transform (DyWT) algorithm was compared with our new Adaptive Gaussian Derivative Filter (AGDF) algorithm for pitch detection of synthesized speech under different noise conditions and signal-to-noise ratios. The results showed that the AGDF outperforms the DyWT pitch detection scheme at low signal-to-noise ratios for different types of noise.

3. An adaptive comb filtering (ACF) algorithm using the AGDF and DyWT pitch detection algorithms has also been demonstrated for the first time. This speech enhancement approach of enhancing the pitch information is known to improve the overall performance of many speech applications including speaker recognition systems. The ACF algorithm was fed pitch information from the AGDF and DyWT pitch detection algorithms and the performance of both methods were compared. The cepstral distance method was used as an objective measure to determine performance of the algorithms. As expected, the both algorithms showed great potential in reducing the distortions due to noise in a corrupted signal. The algorithms showed extremely good performance in the presence of white gausian noise. This confirms the fact that the both AGDF and DyWT pitch detection algorithms performed very well on speech corrupted by white gaussian noise discussed in detail in Chapter 4. In addition, the both algorithms reduced the presence of noise in signals corrupted by babble and colored noise. These results are particularly important for Speaker ID systems which rely heavily on robust cepstral features for improved results.

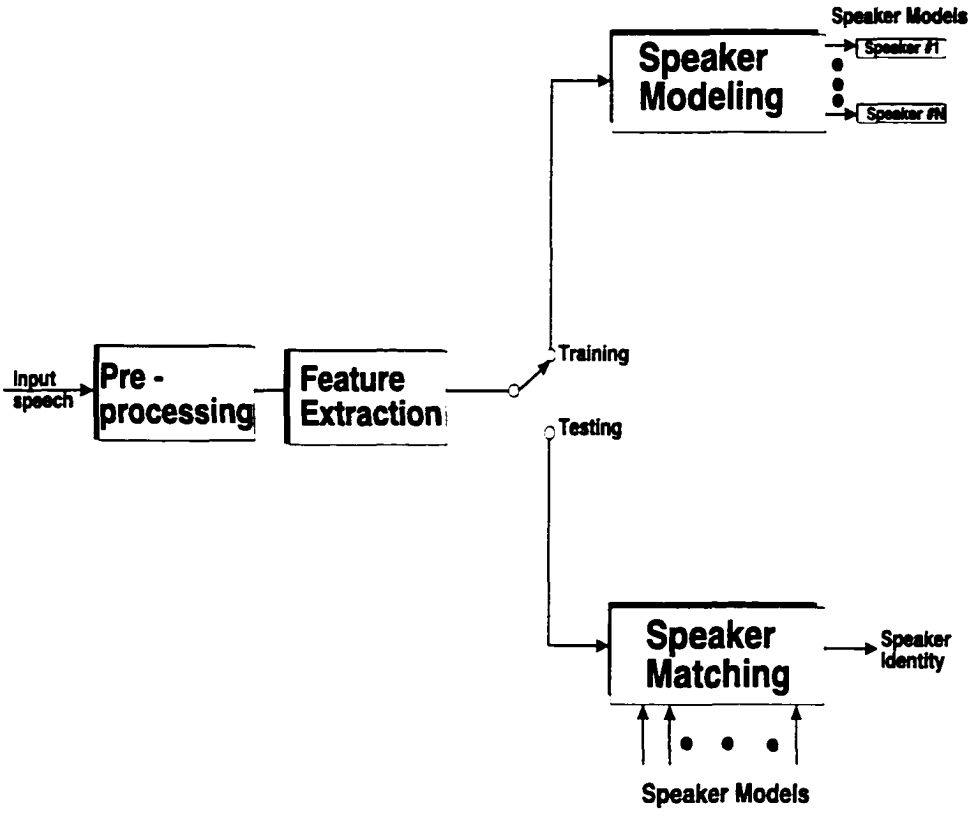


Figure 6.1: The basic structure of a speaker recognition system

6.2 Future Work: New Pitch Based Preprocessors For Speaker Recognition Systems

1 Introduction

Speaker Identification Systems (SIDS) consist of an analysis and pattern recognition phases. The analysis stage consist of using either LP analysis or filter bank theory.

feature extraction (e.g. pitch, cepstrum, etc.) and the pattern recognition stage involves making a decision based on matching speakers' characteristics with existing models. In Figure 6.1, a generic speaker ID system is illustrated.

Unlike the speech recognition problem which is interested in the actual text of what is said by the speaker, the speaker recognition system is only interested in the information speech signal related to the discriminant feature of the speaker. These discriminant features are due to the inherent differences in the articulatory organs (i.e. the structure of the vocal tract, the size of the nasal cavity and vocal cord characteristics) and the manner of speaking of the speaker. The system can be either text-independent (constraint on what is spoken) or text-independent (no constraint on what is spoken).

The feature extractor normalizes the collected data and transforms them into feature space. In feature space, the data is compressed and represented in such an effective way that objects from the same class behave similarly and a clear distinction among objects from different classes exists. The classifier takes the features and performs either a template matching or probabilistic likelihood computation on the features depending on the type of algorithm employed. Prior to the classification stage, the classifier must be trained so that a mapping from the feature to the label of a particular class is established. Since an object is characterized in the classifier by a module or a part of an integrated model, training can also be viewed as a stage of enroll-

ment into the system. This method has been demonstrated to adequately perform speaker recognition task. However, the implicit assumption with this approach is that the training and testing conditions are comparable. However, in the real world the problem of the mismatch conditions arise. This problem has been addressed in detail in the chapter related to the *speaker count determination* problem.

Robust speech techniques have been discussed extensively in [39], in an effort to maintain the performance of a speech processing system under diverse conditions of operation. There are two main strategies which have been shown to mitigate problems that arise due to channel effects and noise. The first strategy aims at making the classifier more robust by compensating for the distortions at the classification stage using statistical approaches. The other strategy, which is of interest to this thesis, deals with making the SIDS more robust in the front-end. One method uses speech enhancement approaches such as Spectral Subtraction, discussed in Chapter 5, so that the features are more representative of clean speech in that noise effects are removed. In Chapter 5, a new speech enhancement approach was developed using the ACF in conjunction with the AGDF and DyWT robust pitch detection methods. In this thesis, we propose to use this method for the first time for improved SIDS performance.

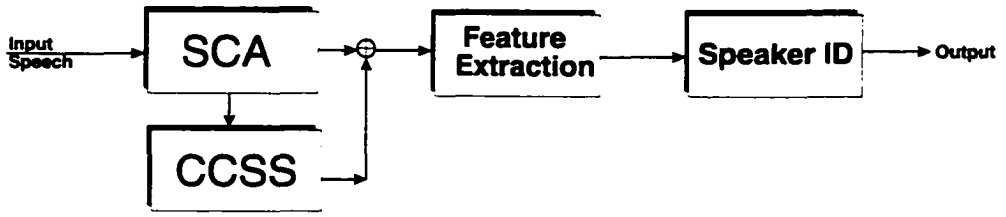


Figure 6.2: The use of the SCA for speaker ID

Figure 6.3: The use of the SEA for speaker ID

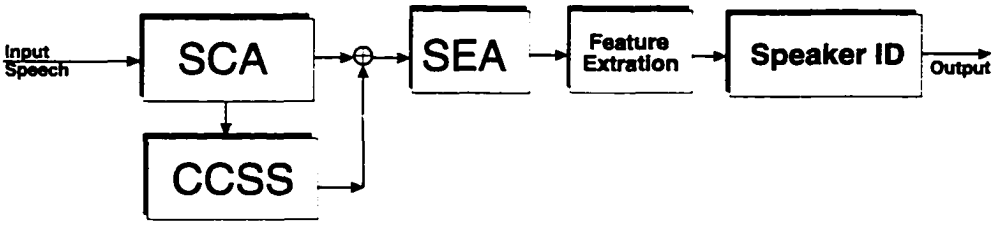


Figure 6.4: A combination of the SCA and SEA for Speaker ID

2 Proposed Algorithms for Robust SpeakerID Systems

In Chapter 3, a new method known as the PPF Speaker Count Algorithm (SCA) was introduced to determine the number of speakers in a frame of speech. This feature showed great potential in solving the *speaker count determination* problem which deals with the automatically identifying frames that have corrupted by cochannel noise or interfering speakers. It is our contention that once these corrupted frames have been labelled or optionally passed through a Cochannel Speaker Separation(CCSS) system, the overall performance of the SIDS should improve with one-speaker speech since it is possible to weight corrupted frames less than non-corrupted frames. Even though the different speakers might be intermittently speaking from frame to frame, as long as the utterance is long enough ($> 300ms$) to extract enough features for a target speaker, the SIDS results should be enhanced. If the interfering speaker happens to also be enrolled in the system, a long term study of the utterance related to signal energy levels can reveal the target speaker over the interfering speaker. A diagram illustrating this proposed algorithm for future study is shown in Fig. 6.2.

In Chapter 4 and 5, an algorithm was developed using either the DyWT or AGDF pitch detection algorithms in conjunction with the ACF Speech Enhancement Algorithm (SEA). It has been shown in [38], that speaker identification results improved tremendously when the voiced frames were enhanced by adaptive comb filtering.

The performance of the speech enhancement is dependent on the pitch information fed to the ACF algorithm. In [55], it was established that the DyWT pitch detector was most robust to noise than all the other traditional pitch detectors. We have subsequently showed that our AGDF algorithm is more robust and accurate than the DyWT algorithm for the determination of pitch period under varying noise conditions. Therefore, the proposed SIDS shown in Fig. 6.3 should experience better performance. It is also possible to add objective distance measures to establish confidence in the results derived in the classifier.

The both algorithms, SEA and SCA can also be combined to give further improved results. The system will be able to address the problem of noise of interfering speakers and at the same time enhance the speech passed by the SCA as shown in Fig. 6.4. Of course, this will be extremely complex and limiting since the pitch prediction algorithm used in the SCA is not very robust to other types of noise. This problem can be alleviated by using the AGDF and DyWT for pitch detection instead of the Pitch Prediction method. A SIDS that is robust to cochannel interference as well as other types of noise will be very much appreciated in the speech processing industry.

6.3 Conclusion

In this thesis, new robust pitch-based techniques to improve speech processing applications were addressed. In Chapter 3, the problem defined as *speaker count determination* was addressed. A new feature known as the Pitch Prediction Feature (PPF) was introduced. When compared with other cepstral features, the PPF gave the best results in identifying speech corrupted by interfering speakers. This feature has many applications for making Cochannel Speaker Separation Systems more practical and, in general, shows potential in solving the cochannel mismatch problem. Secondly, an adaptive gaussian derivative filter (AGDF) has been introduced for the robust determination of the pitch period in speech. The quadratic spline DyWT Wavelet was chosen as a benchmark for comparison since it was shown by Kadambe, et al [55] to outperform all other pitch detection algorithms for low and high pitch speakers in noise. When the AGDF detector was compared with the DyWT pitch detector, it is shown that the AGDF is more robust than the DyWT pitch detector for white noise and babble noise conditions at low SNRs. Finally, the AGDF and DyWT pitch detection algorithms were applied to an adaptive comb filtering scheme for speech enhancement in Chapter 5. Using the cepstral distance as an objective measure, it was demonstrated that both ACF-DyWT and ACF-AGDF algorithms had lowered the cepstral distance of noisy speech. The most dramatic results came wgen the algorithms were used to enhance speech corrupted by white gaussian noise. There was also improved results of speech corrupted with babble and colored noise

at low SNRs.

6.4 Publications

Journal Papers

1. Cochannel speaker count labelling based on cepstral and pitch prediction derived features submitted to Pattern Recognition

2. An Adaptive Gaussian Derivative Filter for Robust Pitch detection of speech. submitted to Signal processing.

3. Adaptive Comb Filtering for speech enhancement using dyadic wavelet transforms. in preparation.

4. Adaptive Comb Filtering for speech enhancement using the Gaussian derivative filter. in preparation.

Conference papers

1. On the use of cepstral and pitch prediction features for speaker count labelling of cochannel speech. accepted in International Conf. on Signal Processing Applications and Technologies.

2. The Use of an Adaptive Gaussian Derivative Filter for Robust Pitch Detection of Noisy Speech", submitted to IASTED International Conf. on Signal and Image Processing, Las Vegas, Nevada, October 28–31, 1998.

Appendix A

Hermite Functions

Hermite functions are are assocated with Hermite polynomials and Gaussian functions. These functions are widely used in quantum physics where they are the eigenfunction solution to the shrodinger equations for a simple linear harmonic oscillator.

The equation for the Hermite functions $h_n(x)$ is given by

$$h_n(x) = H_n(x) \exp(-\frac{x^2}{2}). \tag{A.1}$$

where $n = 0, 1, 2, 3, \cdots$ and $H_n(x)$ is the nth Hermite type H_n polynomial and is defined by

$$H_n(x) = (-1)^n \exp(x^2) \frac{d^n}{dx^n} \exp(-x^2).$$

It is important to note that gaussian terms may be described with offset and spatial

width where the argument x of the exponential terms can be substituted by $(x - x_0)/\sigma$.

The Fourier transform of the $h_n(x)$ functions defined in equation A.1 is given by

$$\mathcal{F}[h_n(x)] = \tilde{h_n}(u) = (i\omega)^n \exp\left(\frac{\omega^2}{2}\right) \tag{A.2}$$

The most general and beneficial property of the hermite function is that it is equivalent in shape in both the time and frequency domain [61]. This means that the spatial and frequency bandwidth are equivalent i.e. $(\Delta x = \Delta\omega)$. In [61], the spatial and frequency bandwidth of the individual Hermite functions is described as $\Delta x = \Delta w = \sqrt{(2n+1)/2}$ where $n = 1, 2, 3, \cdots$. Therefore, the conjoint Uncertainty relationship is linear and given by $\Delta x \Delta\omega = n + 1/2$.

Appendix B

Time-Frequency Transforms

The STFT can be defined as a windowed fourier transform. Basically, the signal is transversed by a shifting window function $\omega(t-\tau)$ and the Fourier transform of the windowed signal is taken. This 'local' Fourier transform is given by

$$STFT_f(w,\tau) = \int_{-\infty}^{\infty} \omega * (t-\tau) f(t) e^{-j\omega t} dt \tag{B.1}$$

The STFT operator has a fixed resolution window in which the time duration of the analysis window is inversely proportional to the bandwidth of the filters. Since each elementary function used in the expansion has the same time and frequency resolution, it means that these filters have poor time and frequency localization. In other words, high frequency resolution results in poor time localization and vice versa.

The fixed window property of the STFT is not desirable since the characteristics of a signal may vary widely. This means that the use of STFT is not ideal because of their poor time and frequency localization property which makes them mathematically deficient in the analysis of many types of signals including speech. In addition. STFTs are inappropriate for observing the fast changes in the speech signal.

Wavelets are a family of functions that can be described as a shifting and scaling of the "mother wavelet" $\psi(t) \in L_2(R)$ expressed as

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}}\psi(\frac{t-b}{a})$$

where $a, b \in R(a \neq 0)$, and the normalization ensures that $\|\psi_{a,b}(t)\| = \|\psi(t)\|$.

The admissability condition states that

$$C_\psi = \int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{\omega} dw < \infty,$$

where $\Psi(\omega)$ is the fourier transform of $\psi(t)$.

The admissabile condition dictates that the fourier transform of the wavelet meet the following conditions:

1.

$$\int_{-\infty}^{\infty} \psi(t)dt = \Psi(0) = 0,$$

This is true since the fourier transform is zero at the origin and the spectrum decays at high frequencies, similarly to a bandpass filter.

2. The normalization of the energy is unity, i.e.

$$\|\psi(t)\|^2 = \int_{-\infty}^{\infty} |\psi(t)|^2 dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |\Psi(\omega)| d\omega = 1.$$

B.1 Properties of the Wavelet Transforms

The Continuous Wavelet Transform is defined by [32] as a function that has a zero mean and satisfies the form

$$CWT_x(b,a) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} \psi^* \left(\frac{t-b}{a} \right) f(t) dt$$

The factor $\frac{1}{\sqrt{a}}$ is used to conserve the norm and $\psi^*(t)$ is the complex conjugate of $\psi(t)$.

Though the CWT has been shown to be an effective tool in speech processing, the CWT is highly redundant and computationally complex. These undisirable properties can be reduced by sampling both the translation parameter 'b' and the scale parameter 'a'. This leads to one of the more popular versions of the wavelet transforms, the Discrete Wavelet Transform (DWT) which is discussed in the following section.

The DWT of a signal f(t) can be expressed by discretizing both the scale parameter 'a' and the translation parameter 'b' in equation B.1. Therefore,

$$DWT_x(nb_0, a_0^{m/2}) = \frac{1}{\sqrt{a_0^m}} \int_{-\infty}^{\infty} \upsilon^* \left(\frac{t - nb_0}{a_0^m} \right) f(t)dt$$

The factor $\frac{1}{\sqrt{a_0^{m/2}}}$ is used to conserve the norm and $\upsilon*(t)$ is the complex conjugate of $\upsilon(t)$. The main properties of the DWT_f are:

1. **Linearity Property** The DWT is linear, whereas the DWT of the DWT is non-linear and introduces cross tems similar to the CWT. If we have a multi-component signal $\hat{x}(t) = \sum_{i=1}^{n} x_i(t)$, then the DWT is given by:

$$DWT_{\hat{x}}(n, m) = \sum_{i=1}^{n} DWT_{x_i}(n, m)$$

However, the energy distribution of the DWT is non linear and is given by

$$\begin{aligned} |DWT_{\hat{x}}(n, m)|^2 &= \sum_{i=1}^{n} |DWT_{x_i}(n, m)|^2 \\ &\quad + 2Re \sum_{k=1}^{m} \sum_{l=k+1}^{n} DWT_{x_k}(n, m) \\ &\quad \times DWT^*_{x_l}(n, m) \end{aligned}$$

where $DWT^*_{x_l}(n, m)$ is the complex conjugate of $DWT_{x_l}(n, m)$. The first term of equation B.2 corresponds to the energy distribution of the DWT of the signal $x_i(t)$. The second summand of the equation is called the cross terms.

2. **Scale invariant Property** Unlike the CWT, the DWT is scale invariant. For example, if we have a signal $x_1(t) = \lambda x \lambda t$, is a scaled version of the signal $x(t)$ then the it can be shown that

$$DWT_{x_1}(n, m) \neq DWT_x(\lambda n, \lambda m)$$

3. **Shift Property** The DWT is time shift variant. This means that the DWT of a time shifted signal is not shifted in time by the same amount. If we have a signal $x_1(t) = x(t - t_1)$ then $DWT_{x_1}(n, m) \neq DWT_x(n - t_1, m)$

4. **Modulation Property** The DWT of a frequency modulated signal $x_1(t) = x(t) \exp j\omega_1 t$ is given by

$$DWT_{x_1}(n, m) = DWT_{x,\dot{G}}(n, m) \exp(jna_0^m b_0 \omega_1)$$

where $\dot{G}(a\omega) = G(a(\omega + \omega_1))$

5. **Reconstruction Property** The signal x(t) can be reconstructed from its DWT i.e.

$$x(t) = \sum_{n,m} DWT_x(n, m) a_0^{-\frac{m}{2}} \phi(a_0^m t - nb_0)$$

Appendix C

Sensitivity Analysis of The Spatial Width of The Gaussian Derivative Filter

The Gaussian Derivative Filter (GDF) can be described by

$$g_D = c_0 h_0(x/\sigma) + c_2 h_2(x/\sigma) \tag{C.1}$$

Where $c_2 < 0$ for all c_2. The GDF can be normalized by making the coefficient $c_0 + c_2 = 1$. Then we can rewrite equation C.1 as

$$g_D = c_0(h_0(x/\sigma) + c_2 h_2(x/\sigma)) - h_2(x/\sigma) \tag{C.2}$$

where

$$h_0(x/\sigma) = \frac{1}{\sigma\sqrt{\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{C.3}$$

and

$$h_2(x/\sigma) = \frac{1}{\sqrt{8\pi\sigma^2}}\left(-\exp\left(-\frac{x^2}{2\sigma^2}\right) + \frac{x^2}{\sigma^2}\exp\left(-\frac{x^2}{2\sigma^2}\right)\right) \tag{C.4}$$

If equation C.3 and C.4 are substituted into equation C.2 then

$$g_D(f) = A(B + x^2)\exp\left(-ax^2\right) \tag{C.5}$$

where $A = \frac{1}{\sqrt{8\pi\sigma^2}}, B = 63c_0 + 1, C = \frac{c_0 - 1}{\sigma^2}$ and $a = \frac{1}{2\sigma^2}$.

The spatial bandwidth of a function f(x) can be calculated using the following relationship:

$$(\Delta x)^2 = \frac{\int_{-\infty}^{\infty}(x - \mu_x)^2|f(x)|^2dx}{\int_{-\infty}^{\infty}|f(x)|^2dx} \tag{C.6}$$

Based on the properties of the gaussian derivative function, μ_x, the mean of the function is zero.

Using equation C.5 in equation C.6 and keeping in mind that the

$$\int_0^{\infty} x^n \exp\left(-ax^2\right) = \frac{\Gamma((n+1)/2)}{a^((n+1)/2)}.$$

then the spatial width can be derived as

$$\Delta x = \tau^{1/2}\sigma \tag{C.7}$$

where

$$\tau = \frac{1 + 3\gamma + 5\gamma^2}{2 + 2\gamma + 1.5\gamma^2} \tag{C.8}$$

and

$$\gamma = \frac{c_0 - 1}{63c_0 + 1} \tag{C.9}$$

It is now possible to find the effects of the parameters c_0 and σ on the spatial bandwidth by performing a sensitivity analysis. The sensitivity of a parameter on a function is described as the ratio of the percentage change of the function over the percentage change of the paramter. From [54], the following equations has been adapted for this purpose. For the spatial bandwidth expressed in equation C.7, we can be express the following sensitiviy expressions:

$$S_{\sigma}^{\Delta x} = \frac{\partial \Delta x}{\partial \sigma}\frac{\sigma}{\Delta x} \tag{C.10}$$

and

$$S_{c_0}^{\Delta x} = \frac{\partial \Delta x}{\partial c_0}\frac{\dot{c_0}}{\Delta x} \tag{C.11}$$

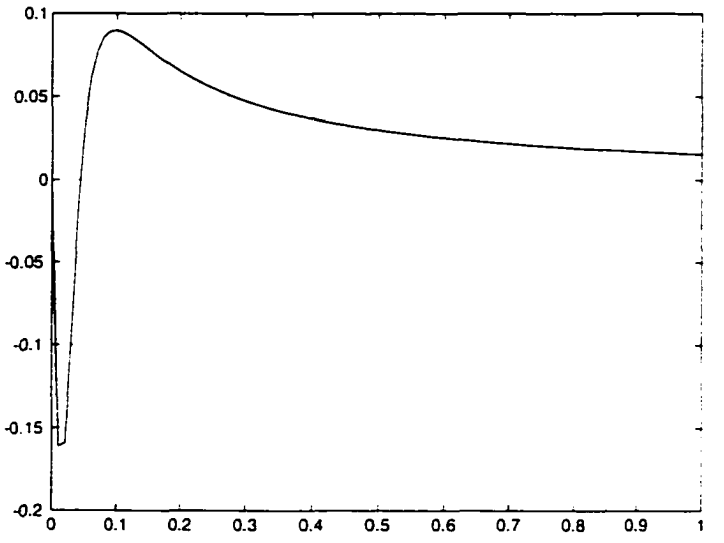


Figure C.1: The sensitivity of the c_0 on the spatial bandwidth

From equation C.10, it can be determined that there is a unity relationship between the ratio of percentage change in the spatial bandwith to the scaling parameter σ i.e. $S_{\sigma}^{\Delta x} = 1$. However, in terms of the c_0 parameter, the relationship is less obvious. From equation C.11, the sensitivity can be expressed as

$$S_{c_0}^{\Delta x} = \frac{\partial \Delta x}{\partial \tau} \frac{\partial \dot{\tau}}{\partial \dot{\gamma}} \frac{\partial \dot{\gamma}}{\partial c_0} \tag{C.12}$$

From this equation the sensitivity has been derived as

$$S_{c_0}^{\Delta x} = \frac{8 + 34\gamma + 11\gamma^2}{(4 + 4\gamma + 3\gamma^2)(1 + 3\gamma + 5\gamma^2)} \frac{32\dot{c}_0}{(63c_0 + 1)^2} \tag{C.13}$$

This sensitivity relationship is illustrated in Figure C.1. Therefore, $max|S_{\sigma}^{c_0}| = .16$.

Bibliography

[1] Michael Savic, E. Acosta and Sunil K. Gupta, "An Automatic Langauge Identification System", *ICASSP 91* pp.817-820 1991.

[2] J.L. Flanagan, C. H. Coker, L. R. Rabiner, R. W. Shafer, and N. Umeda. Syntehetic voices for computers. *IEEE Spectrum, Vol. 7 No. 10 pages. 22-45. October 1970.*

[3] A. V. Oppenheim, R. W.Shafer, Discrete-Time Signal Processing, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[4] L. R. Rabiner and B. Juang, Fundamentals of Speech Recognition. Prentice-Hall. 1993.

[5] R. Frazier, Siamak Samsam, L. Braida, A. V. Oppenheim, "Enhancement of Speech By Adaptive Filtering",IEEE Int. Conf. on Acoustics. Speech. and Signal Processing,pp.251-253, April 1976.

[6] O. Fujimura, "Analysis of nasals consonants". Journal of the Acoustical Society of America. vol. 34, pp. 1865-1875. Dec. 1962.

[7] Jae S. Lim, A. V. Oppenheim, L. D. Braida, " Evalulation of an Adaptive Comb Filtering Method for Enhancing Speech Degraded by White Noise Addition". IEEE Transactions on Acoustics, Speech, and Signal, Vol. ASSP-26, No.5,pp. 354-358, August 1978.

[8] R. P. Ramachandran and P. Kabal, "Pitch prediction filters in speech coding". IEEE Trans. on Acoust., Speech and Signal Proc., vol. 37, pp. 467-478. April 1989.

[9] J. Makhoul, S. Roucos and H. Gish, "Vector quantization in speech coding". Proc. IEEE, vol. 73, pp. 1551-1588, Nov. 1985.

[10] A. Gersho and R. M. Gray, Vector Quantization and Signal Compression. Kluwer Academic Publishers, 1991.

[11] A. Gersho, "Asymptotically optimal block quantization". IEEE Trans. on Infor. Theory, vol. IT-25, pp. 373-380, July 1979.

[12] S. G. Mallat and S. Zhong, "Complete signal representation with multiscale edges," tech. rep. RRT-483-RR-219, Courant Inst. of Math. Sci., Dec. 1989

[13] W. Hess, Pitch determination of speech signals: algorithms and devices. Berlin, West Germany: Springer Verlag, 1983

[14] K. R. Scherer, "Speech and emotional states," Speech evaluation in psychiatry.(G. J. K Darby, ed.), New York, NY: Grune and Stratton, 1981.

[15] A.E. Rosenberg and M. R. Sambur, "New techniques for automatic speaker verification," IEEE transaction on Acoustics, Speech and Signal Processing, vol. ASSP-23, pp.169-176, Apr 1975.

[16] A.E. Rosenberg, "Effect of Glottal Pulse Shape on the Quality of Natural Vowels," J. Acoust. Soc. Am., vol.2, pp.583-590, Feb. 1971.

[17] Javier Ortega-Garcia and Joaguin Gonzalez-Rodriguez, "Overview Of Speech Enhancement Techniques For Automatic Speaker Recognition", ICSLP 96

[18] J. J. Dubnowski, R. W. Shafer, L. R. Rabiner, "Real-time digital hardware pitch detector," IEEE Transactions on Acoustics, Speech and Signal Processing., vol. ASSP-24, pp. 2-8, Feb 1976.

[19] C. J. Bristow and F. Fallside, "An autocorrelation pitch detector with error correction," in Proceedings of the International Conference on ASSP, pp. 184-187.1982.

[20] B. Gold and L. R. Rabiner, "Parallel processing techniques for estimating pitch periods of speech in the time-domain," Journal of the Acoustical Society of America, vol. 46, pp. 442-448, Aug. 1969

[21] A. M. Noll, "Cepstrum pitch determination," Journal of the Acoustical Society of America, vol. 47, pp. 634-648, Feb.1967.

[22] H. W. Strube, "Determination of the instant of glottal closure from the speech wave," Journal of the Acoustical Society of America, vol. 56, pp. 1625-1629, Nov. 1974.

[23] T. V. Ananthapadmanabha and B. Yegnanarayana, "Epoch extraction of voiced speech," IEEE Trans. Acous., Speech , Signal Processing, vol. 23, pp. 562-570, Dec. 1975.

[24] S. Kadambe and P. Srinivasan, "Text-independent Speaker Identification based on Adaptive Wavelets,"SPIE society of photo-optical Instrumentation, vol. 2242,pp. 669-677, 1994.

[25] "Separation of speech from interferring speech by means of harmonic selection"Journal of the Acoustical Society of America, vol. 60,No.4,pp. 911-918, 1976.

[26] Ma93 C. Ma, Y. Kamp and L. F. Willems, "Robust signal selection for linear prediction analysis of voiced speech," Speech Communication, vol. 12, no. 2, pp. 69–81, June 1993.

[27] R. J. Mammone, Computational Methods of Signal Recovery and Recognition, John Wiley & Sons, Inc., 1992.

[28] F. Itakura, "Line spectrum representation of linear predictive coefficients," Jour. Acoust. Soc. of Amer., vol. 57, no. 1, pp. S35, 1975.

[29] K. R. Farrell, R. J. Mammone and K. T. Assaleh, "Speaker recognition using neural networks versus conventional classifiers", IEEE Trans. on Speech and Audio Proc., vol. 2, pp. 194–205, Jan. 1994.

[30] T. K. Ho, J. J. Hull and S. N. Srihari, "Decision combination in multiple classifier systems," IEEE Trans. on Patt. Anal. and Mach. Intell., vol. 16, no. 1, pp. 66–75, 1994.

[31] K. R. Farrell and R. J. Mammone, "Data fusion techniques for speaker recognition", in Modern Methods of Speech Processing, edited by R. P. Ramachandran and R. J. Mammone, Kluwer Academic Publishers, 1995. Pattern Recognition vol. 27, pp.1451-1461, Nov. 1994.

[32] Martin Vetterli and Jelena Kavacevic, Wavelets and Subband coding. New Jersey, USA: Prentice Hall, Inc, 1995.

[33] G. Fant, Acoustic Theory of Speech Production, The Hague: Mouton, 1970.

[34] Y. Linde, A. Buzo and R. M. Gray, "An algorithm for vector quantizer design", IEEE Trans. on Comm., vol. COM-28, pp. 84–95, Jan. 1980.

[35] W. Hock, and K. Schittowski, "A Comparative Performance Evaluation of 27 Nonlinear Programming Codes", Computing Vol.30, pp.335, 1983.

[36] K. T. Assaleh, R. J. Mammone, M. G. Rahim, J. L. Flanagan,"Speech Recognition Using the Modulation Models."Proc. IEEE Int. Conf. Acoust., Speech. Signal Processing, vol. 2, pp. 664-667, April, 1993.

[37] A. E. Rosenberg and F. K. Soong, "Evaluation of a vector quantization talker recognition system in text independent and text dependent modes", Comp. Speech and Lang., vol. 22, pp. 143–157, 1987.

[38] *Ramalho M., "The pitch mode modulation model and its application in speech processing," in Modern Methods of Speech Processing, edited by R. P. Ramachandran and R. J. Mammone, Boston, MA, Kluwer, 1995.*

[39] *J. H. L. Hansen, R. J. Mammone and S. Young, editors. IEEE Transactions on Speech and Audio Processing, October 1994.*

[40] *A. Sankar and R. J. Mammone, "Growing and pruning neural tree networks". IEEE Trans. on Computers, vol. C-42, pp. 221-229, March 1993.*

[41] *L. R. Rabiner and R. W. Schafer, Digital Processing of Speech Signals, Prentice-Hall, 1978.*

[42] *F. K. Soong and B. H. Juang, "Line spectrum pair (LSP) and speech data compression", IEEE Int. Conf. on Acoust., Speech and Sig. Proc., San Diego, California, pp. 1.10.1-1.10.4., March 1984.*

[43] *B. S. Atal, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification." J. Acoust. Soc. Am., vol. 55, pp. 1304-1312, June 1974.*

[44] *S. S. Stevens, "Critical bandwidth in loudness summation." J. Acoust. Soc. Am., vol. 29, pp. 548-557, 1957.*

[45] *S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences." IEEE Trans. on Acoust., Speech and Signal Proc., vol. ASSP-28, pp. 357-366, August 1980.*

[46] *R. O. Duda and P. E. Hart, Pattern Classification and Scene Analysis, John Wiley and Sons, 1973.*

[47] *R. J. Schalkoff, Digital Image Processing and Computer Vision, John Wiley and Sons, 1989.*

[48] *G. Deng and L. W. Cahill, "An adaptive Gaussian filter for noise reduction and edge detection," IEEE Nuclear Science Symposium and Medical Imaging Conf., pp. 1615-1619, 1994.*

[49] *T. V. Ananthapadmanabha and B. Yegnanarayana, "Epoch extraction of voiced speech," IEEE Trans. Acous., Speech , Signal Processing, vol. 23, pp. 562-570, Dec. 1975.*

[50] *Basu M., "Gaussian Derivative model for Edge Detection," Pattern Recognition 27:11:1451-1461, 1994.*

[51] *Chin R. T. and Yeh C. L.. "Quantitive evaluation of some edge preserving noise smoothing techniques." Comput. Vision, Graphics, Image Processing 23:67-91. 1983.*

[52] *Davis L. S. and Rosenfeld A.,"Noise cleaning by iterated local averaging." IEEE Transaction on Systems, Man and Cybernetics,SMC-8:705-710. 1978.*

[53] *Davis S.B.. " Acoustic characteristics of laryngeal pathology,"in Speech evaluation in medicine, New York, NY, Grune and Stratton. 1981.*

[54] *Dorf R. C. and Bishop R. H.. " Modern Control Systems" Addison-Wesley. 1998.*

[55] *Kadambe S. and Bourdreaux-Bartels G. F.. "Applications of the Wavelet Transform for Pitch Detection of Speech Signals." IEEE Trans. on Information Theory 38:917-924, Mar. 1992.*

[56] *D. Marr and E. Hildreth, "Theory of edge detection." Proceedings of the Royal Society of London. B:207:187-217, 1980.*

[57] *Medan Y., Yair E. and Chazan D.."Super resolution pitch determination of speech signals", IEEE Transactions On Signal Processing, 39:1:40-48. 1991.*

[58] *Saint-Marc P., Chen J. and Medioni G., "Adaptive Smoothing:A General tool for Early Vision," IEEE Transaction On Pattern Analysis and Machine Intelligence, 13(6):514-529, 1991.*

[59] *Strube H. W., "Determination of the instant of glottal closure from the speech wave." Journal of the Acoustical Society of America, 56:1625-1629, 1974.*

[60] *Young R.. "The Gaussian Derivative Model For Spatial Vision: I. Retinal Mechanism." Spatial Vision, 2(4):273-293, 1987*

[61] *Young R.. "Oh say, can you see? The physiology of vision", SPIE Human Vision, Visual Processing, and Digital Display II Vol. 1453, pp. 92-122. 1991.*

[62] *M. A. Zissman, C. J. Weinstein and L. D. Braida, "Automatic talker activity labelling for co-channel talker inteference suppression",IEEE Int. Conf. on Acoust., Speech and Sig. Proc., Albuquerque, New Mexico, pp. 813-816. April 1990.*

[63] *R. P. Ramachandran and P. Kabal, "Pitch prediction filters in speech coding", IEEE Trans. on Acoust., Speech and Signal Proc., vol. 37, pp. 467-478, April 1989.*

[64] *Y. Linde, A. Buzo and R. M. Gray, "An algorithm for vector quantizer design", IEEE Trans. on Comm., vol. COM-28, pp. 84-95, Jan. 1980. J. Acoust. Soc. Am., vol. 55, pp. 1304-1312, June 1974.*

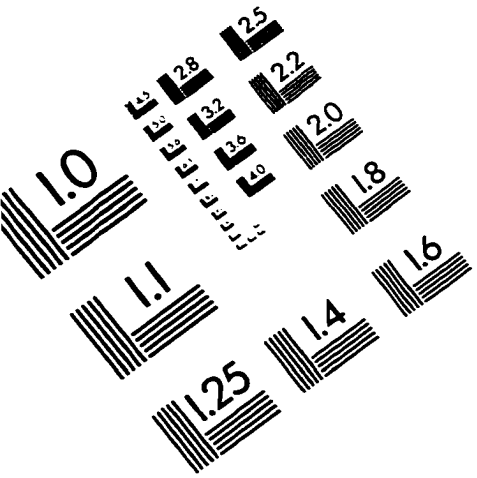
[65] L. R. Rabiner and B. H. Juang. Fundamentals of Speech Recognition. Prentice-Hall. 1993.

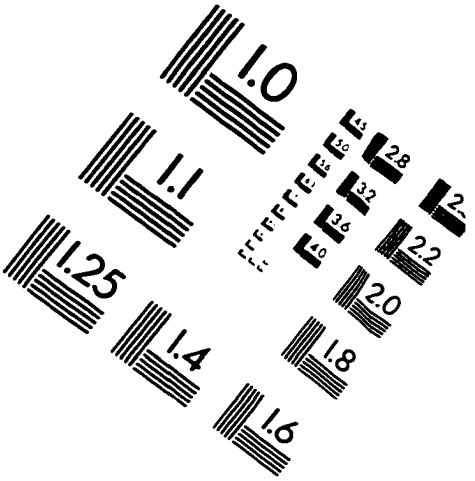
[66] B. H. Juang, L. R. Rabiner and J. G. Wilpon, "On the use of bandpass liftering in speech recognition", IEEE Trans. on Acoust., Speech and Sig. Proc., vol. ASSP-35. pp. 947-954, July 1987.

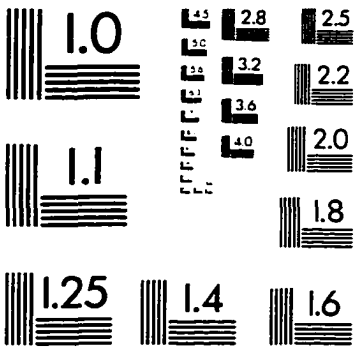
[67] K. T. Assaleh and R. J. Mammone, "New LP-derived features for speaker identification", IEEE Trans. on Speech and Audio Proc., vol. 2, pp. 630-638. Oct. 1994.

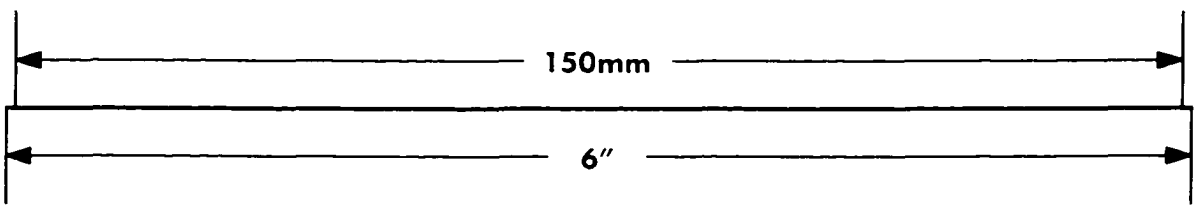
[68] M. S. Zilovic, R. P. Ramachandran and R. J. Mammone, "A fast algorithm for finding the adaptive component weighted cepstrum for speaker recognition". IEEE Trans. on Speech and Audio Proc., vol. 5. pp. 84-86. Jan. 1997.

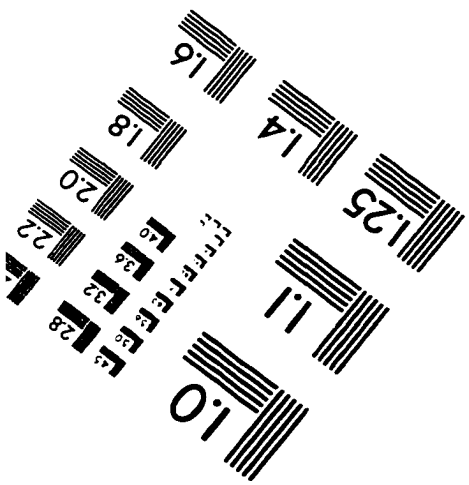
IMAGE EVALUATION TEST TARGET (QA-3)

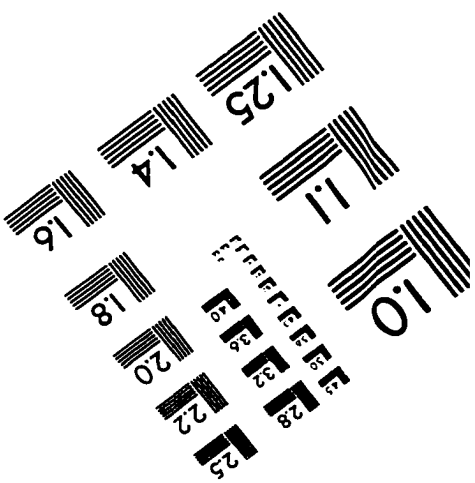












APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989