Speech Endpoint Detection: An Image Segmentation Approach

by

Nesma Faris

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2013

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Speech Endpoint Detection, also known as Speech Segmentation, is an unsolved problem in speech processing that affects numerous applications including robust speech recognition. This task is not as trivial as it appears, and most of the existing algorithms degrade at low signal-to-noise ratios (SNRs). Most of the previous research approaches have focused on the development of robust algorithms with special attention being paid to the derivation and study of noise robust features and decision rules. This research tackles the endpoint detection problem in a different way, and proposes a novel speech endpoint detection algorithm which has been derived from Chan-Vese algorithm for image segmentation. The proposed algorithm has the ability to fuse multi features extracted from the speech signal to enhance the detection accuracy. The algorithm performance has been evaluated and compared to two widely used speech detection algorithms under various noise environments with SNR levels ranging from 0 dB to 30 dB. Furthermore, the proposed algorithm has also been applied to different types of American English phonemes. The experiments show that, even under conditions of severe noise contamination, the proposed algorithm is more efficient as compared to the reference algorithms.

Acknowledgements

First and foremost, I thank God for granting me the strength and knowledge to complete this work.

I would like to express my deep and sincere gratitude and appreciation to my supervisors Dr. Otman Basir and Dr. Oleg Michailovich for their guidance, extensive assistance, critical insight, and patience. The opportunity to work with them has been tremendously rewarding.

I am very grateful to the Libyan Ministry of Education for providing the financial support throughout my MASc. program.

Last but not least, I owe a world of thanks to my best friend and my husband, Wail Menesi, for his continuous support and encouragement during these years. Without his help, I would have never made it.

*To the soul of my father (**Fathi Faris**), for his unforgettable support, guide, and love*

*To my mother **(Nabila Ben Mussa)**, for her love and outstanding support*

Table of Contents

List of Tables viii

List of Figures ix

1 Introduction 1

1.1 Introduction........ 1

1.2 Research Motivation and Objective........ 3

1.3 Thesis Organization........ 4

2 Background and Literature Review 6

2.1 Introduction........ 6

2.2 Speech Production........ 6

2.3 Phonetic Classification of Speech........ 8

2.4 Literature Review........ 10

3 Speech Endpoint Detection: An Image Segmentation Approach 22

3.1 Introduction........ 22

3.2 Speech Enhancement........ 23

3.3 Signal Preprocessing........ 26

3.3.1 Pre-emphasis........ 26

3.3.2 Framing and Windowing........ 28

3.4 Feature Extraction........ 30

3.4.1 Log-energy........ 31

3.4.2 Perceptual Linear Predictive Coefficients........ 33

3.4.3 Mel-Frequency Cepstral Coefficients........ 37

3.5 Speech Detection........ 41

3.5.1 Problem Description........ 42

3.5.2 Regularizations of the Heaviside and Dirac Delta Function........ 45

3.5.3 Fusion of Different Features........ 46

3.5.4 Summary of the Proposed Algorithm........ 48

4 Experimental Validation 49

4.1 Performance Analysis........ 49

4.1.1 Support Database........ 49

4.1.2 Experimental Results........ 50

4.2 Multi-phoneme Comparison between Algorithms........ 55

4.3 Summary........ 77

5 Conclusion and Future Work 78

5.1 Conclusion........ 78

5.2 Future Work........ 78

References 80

List of Tables

4.1 MSE for various algorithms under comparison (SNR = Inf dB)............... 50

4.2 MSE per phoneme, in case of clean speech…………………………………….. 57

4.3 MSE per phoneme type, in case of white noise (0dB ≤SNR<10dB)…… 59

4.4 MSE per phoneme type, in case of white noise (10dB ≤SNR<20dB)…. 61

4.5 MSE per phoneme type, in case of white noise (20dB ≤SNR≤30dB)…. 63

4.6 MSE per phoneme type, in case of pink noise (0dB≤ SNR<10dB)……... 65

4.7 MSE per phoneme type, in case of pink noise (10dB ≤SNR<20dB)…… 67

4.8 MSE per phoneme type, in case of pink noise (20dB ≤SNR≤30dB)…… 69

4.9 MSE per phoneme type, in case of car noise (0dB ≤SNR<10dB)………. 71

4.10 MSE per phoneme type, in case of car noise (10dB≤ SNR<20dB)……... 73

4.11 MSE per phoneme type, in case of car noise (20dB ≤SNR≤30dB)……... 75

List of Figures

1.1 Block Diagram of Speech Recognizer using an Explicit Endpoint Detector [2]………………………………………………………………………………… 1

2.1 Schematic View of the Human Vocal Mechanism [8]…………………………… 7

2.2 Schematic Representation of the Complete Physiological Mechanism of Speech Production [3]……………………………………………………………………. 8

2.3 Classifications of American English Phonemes into Broad Sound Classes [10]……………………………………………………………………………………… 9

3.1 Block Diagram of the Proposed Algorithm……………………………………….. 23

3.2 Block Diagram of the Multiband Spectral Subtraction Algorithm [10]..... 26

3.3 Pre-emphasis Filter Frequency Response………………………………………... 27

3.4 Framing and Overlapping……………………………………………………………... 28

3.5 Hamming Window Function for M=60…………………………………………….. 30

3.6 Waveform of the Word ”Hot”(Top Plot), and Corresponding Energy Curve (Bottom Plot)……………………………………………………………………. 33

3.7 Block Diagram of Perceptual Linear Predictive (PLP) Analysis of Speech……………………………………………………………………………………… 34

3.8 The Bark Scale………………………………………………………………………….. 35

3.9 Waveform of the Word ”Hot”(Top Plot), and Corresponding PLP Curve (Bottom Plot).. 37

3.10 The Mel Scale…………………………………………………………………………….. 38

3.11 Triangular Filters Used to Compute MFCC……………………………………….. 39

3.12 Block Diagram of MFCC Computation Process…………………………………... 40

3.13 Waveform of the Word ”Hot”(Top Plot), and Corresponding MFCC Curve (Bottom Plot)…………………………………………………………………….. 41

3.14 The Distance Function (Top Sketch), and its Corresponding Signed Distance Function (Bottom Sketch)……………………………………………… 43

3.15 Regularized Heaviside and Dirac Delta Functions……………………………. 47

4.1 MSE of the proposed algorithm against the two reference algorithms with white noise added…………………………………………………………….... 52

4.2 MSE of the proposed algorithm against the two reference algorithms with pink noise added………………………………………………………………. 53

4.3 MSE of the proposed algorithm against the two reference algorithms with car noise added………………………………………………………………….. 54

4.4 MSE per phoneme type, in case of clean speech……………………………….. 58

4.5 MSE per phoneme type, in case of white noise (0dB ≤SNR<10dB)……... 60

4.6 MSE per phoneme type, in case of white noise (10dB ≤SNR<20dB)…… 62

4.7 MSE per phoneme type, in case of white noise (20dB≤ SNR≤30dB)...... 64

4.8 MSE per phoneme type, in case of pink noise (0dB≤ SNR<10dB)........... 66

4.9 MSE per phoneme type, in case of pink noise (10dB≤ SNR<20dB).......... 68

4.10 MSE per phoneme type, in case of pink noise (20dB ≤SNR≤30dB)......... 70

4.11 MSE per phoneme type, in case of car noise (0dB ≤SNR<10dB).............. 72

4.12 MSE per phoneme type, in case of car noise (10dB ≤SNR<20dB)............ 74

4.13 MSE per phoneme type, in case of car noise (20dB ≤SNR≤30dB)............. 76

Chapter 1

Introduction

1.1 Introduction

Speech endpoint detection (also known as *boundary detection*, *voice activity detection*, or *speech segmentation*) is the process of detecting the onset and the terminus of speech utterance and exclusion of the non-speech segments by digital processing technology. Endpoint detection constitutes an indispensable part of numerous applications, such as speech analysis, synthesis and recognition. The study in [1] has shown that the recognition performance has a close relation to the accuracy of endpoint detection. More than half of speech recognition errors were caused by incorrect endpoint detection even in quiet environment. Furthermore, higher detection rates can help to identify and reject background noise, which can in turn reduce the time complexity of speech recognition as well as improve the performance of speech recognition system.

Three types of endpoint detection for speech recognition schemes are currently available: explicit, implicit, and hybrid [2]. Thus, as opposed to the implicit schemes, explicit schemes consist of a separate and independent endpoint detection stage prior to the speech recognition stage. On the other hand, in implicit schemes, the endpoint detection is integrated into the recognition stage. The hybrid schemes essentially combine both explicit and implicit approaches. In [2], it was also indicated that sophisticated explicit endpoint detection schemes usually outperform the other two approaches. The general block diagram of a speech recognizer using an explicit endpoint detector is shown in Fig. 1.1

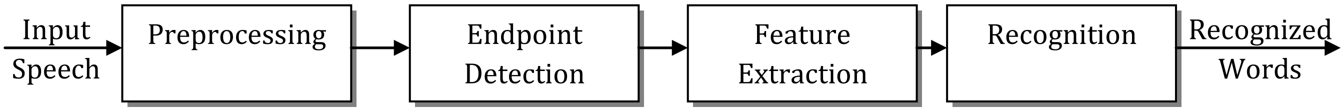


Figure 1.1: Block Diagram of Speech Recognizer using an Explicit Endpoint Detector [2].

The need for speech detection also arises in other applications, including:

- **Analog Telecommunication**: In analog multichannel transmission systems, a technique called *Time-Assignment Speech Interpolation* (TASI) is often used to take advantage of the channel idle time by detecting the presence of a talker's speech and engaging an unused channel only when speech is detected. This allows a substantial improvement in the efficiency (and throughput) of customer services [3].
- **Digital Communication**: Accurate endpoint detection is used in many digital communication systems during discontinuous transmission to optimize power consumption and to minimize the average bit rate, thereby improving the overall encoding quality of the speech [4].

Endpoint detection is a challenging problem. There are numerous obstacles that render endpoint detection difficult. One particular class of such obstacles is those attributed to the speaker and the manner of producing the speech. For example, during articulation, the speaker often produces sound artifacts, including lip smacks, heavy breathing and mouth clicks [3].

Another common factor that makes reliable speech endpoint detection difficult is the environmental conditions in which the speech is produced. An ideal environment for acquiring speech signals is in a quiet room with no acoustic reverberations and/or noise clutter. Unfortunately, such an ideal environment is not always realizable [3]. For example, non-stationary sounds (such as a door slam, a car horn, or even speech interference by a radio, TV or background conversation) may occur during the speech recording. Some of such interfering signals possess as much speech-like features as that of the desired speech signal itself, making accurate endpoint detection a non-trivial problem to solve.

An additional source of signal degradation is the distortion introduced by the transmission system over which the speech signals are sent. Factors like cross-talk, inter-modulation distortion, and various types of tonal interference arise to various degrees in the communication channels [3].

Difficulties in endpoint detection arise not only from the different types of noise present in the recording, but also from the vocabulary words themselves. Some phonemes or sounds have very low energy when compared to the vowel portion of the speech, and as a result, they are interpreted as background noise [5]. Among such phonemes, e.g., the weak fricative /f/, weak plosives /p/or /t/ or nasals such as /n/ at the end.

Among various endpoint detection approaches, energy-based methods are the most widely used. The basic idea of using energy signature to detect endpoints is that, for sufficiently high SNR values, the local energy (or power) of a speech signal is indicative of the utterance rather than of silence. In these methods, a fixed-length window is "slid" over the duration of the input utterance, followed by local energies computation of the signal within time window. By continuously monitoring the local energies, the starting point can be found as the one at which the latter exceed a pre-defined threshold. Similarly, the ending point can be located when the local energies fall below some ending threshold [6].

An efficient endpoint detection algorithm should be accurate, robust, and self-adaptive. Robustness means that the algorithm should be reliable in different noise conditions. Most of the recent endpoint detection methods (such as the methods which are based on short-term energy signature) demonstrate good performance at relatively high SNRs. Unfortunately, these methods become much less reliable as the SNR values decrease [7].

1.2 Research Motivation and Objective

The recognition of isolated words has many potential applications, such as remote data entry via voice commands. Naturally, such systems rely heavily on endpoint detection stage for identifying the speech fragments of a given audio signal. The precise estimation of the endpoints of speech has a significant and direct impact on the performance of the recognizer, since it can potentially increase the recognition accuracy and reduce computation complexity.

However, many words associated with digits as well as those having CVC (Consonant-Vowel-Consonant) composition are characterized by low-energy onsets and “tails”, which make them particularly difficult to accurately determine their actual endpoints. For example, for the digit “eight,” the endpoint detection system could easily miss the final weak portion “t”, especially when the utterance is contaminated with noise. As such, the detection of the weak points of an utterance in the presence of background noise has been considered as a challenging problem which has attracted many researchers trying to develop effective techniques for its solution. Currently, the research on endpoint detection is considered to be a hot topic. However, these research studies are mostly data-specific, and unfortunately, no globally accepted or widely used approach has been proposed so far.

Therefore, the main objective of this research is to develop a robust endpoint detection algorithm for isolated words, which will be efficient, and capable of performing reliably under various noise conditions. The performance of the proposed algorithm will be evaluated for different types of American English phonemes including the weak consonants which are difficult to detect using conventional endpoint detection methods. In addition, the experimental results of the proposed algorithm will be compared to two widely-used endpoint detection algorithms under three different types of noise with SNR values ranging from 0dB to 30 dB.

1.3 Thesis Organization

This thesis is composed of five chapters. Chapter 1 provides an introduction to this thesis. Chapter 2 reviews state-of-the art of endpoint detection algorithms and provides a brief description of the speech production process and the classification of American English phonemes. In Chapter 3, the overall architecture of the proposed endpoint detection algorithm is presented. Chapter 4 presents a series of experiments conducted to evaluate the effectiveness of the proposed algorithm by comparing its performance to two reference algorithms used extensively in the field of endpoint detection. Finally, Chapter 5 summarizes the contributions of this thesis and highlights the directions of future research.

Chapter 2

Background and Literature Review

2.1 Introduction

This chapter reviews state-of-the-art of what has been done over the years to solve the problem of endpoint detection using various features and techniques. A number of speech endpoint detection methods has been reported in the literature. This includes short-time Energy [2], Short-time Zero-crossing Rate [5], Entropy [13], Mel-Frequency Cepstrum Coefficient (MFCC) [7], Hidden Markov Models (HMM) [18], Wavelet Transform technology [22], etc. Before exploring these methods, a brief overview of speech production process and phonetic classification is represented.

2.2 Speech Production

The speech waveform is an acoustic sound pressure wave that originates from voluntary movements of anatomical structures which make up the human speech production system [8]. A schematic view of the human vocal mechanism is shown in Fig. 2.1. Air enters the lungs via the normal breathing mechanism. As air is expelled from the lungs through the trachea, the tensed vocal cords within the larynx are forced to vibrate (in the mode of a relaxation oscillator). The air flow is chopped into quasi-periodic pulses which are then modulated in frequency as they pass through the pharynx (the throat cavity), the mouth cavity, and possibly the nasal cavity. Depending on the positions of various articulators (i.e., jaw, tongue, velum, lips, mouth), different sounds are produced [3].

A simplified representation of the complete physiological mechanism for creating speech is shown in Fig. 2.2. The lungs and the associated muscles act as the source of air for exciting the vocal mechanism. The muscle force pushes air out of the lungs and through the bronchi and trachea. When the vocal cords are tensed, the air flow causes them to vibrate, producing so-called voiced speech sounds [3]. When the vocal cords are relaxed, in order to produce a sound, the air flow must pass through a constriction in the vocal tract and thereby become turbulent, producing so-called unvoiced sounds. Alternatively, it can build up pressure behind a point of total closure within the vocal tract, and when the closure is opened, the pressure is suddenly and abruptly released, causing a brief transient sound [3].

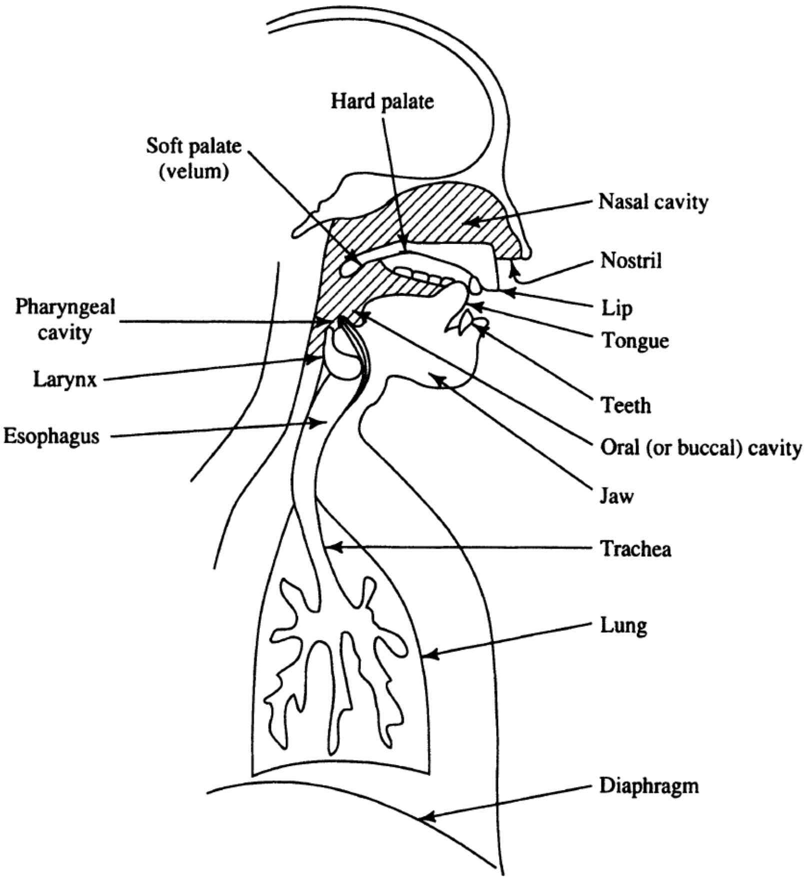


Figure 2.1: Schematic View of the Human Vocal Mechanism [8].

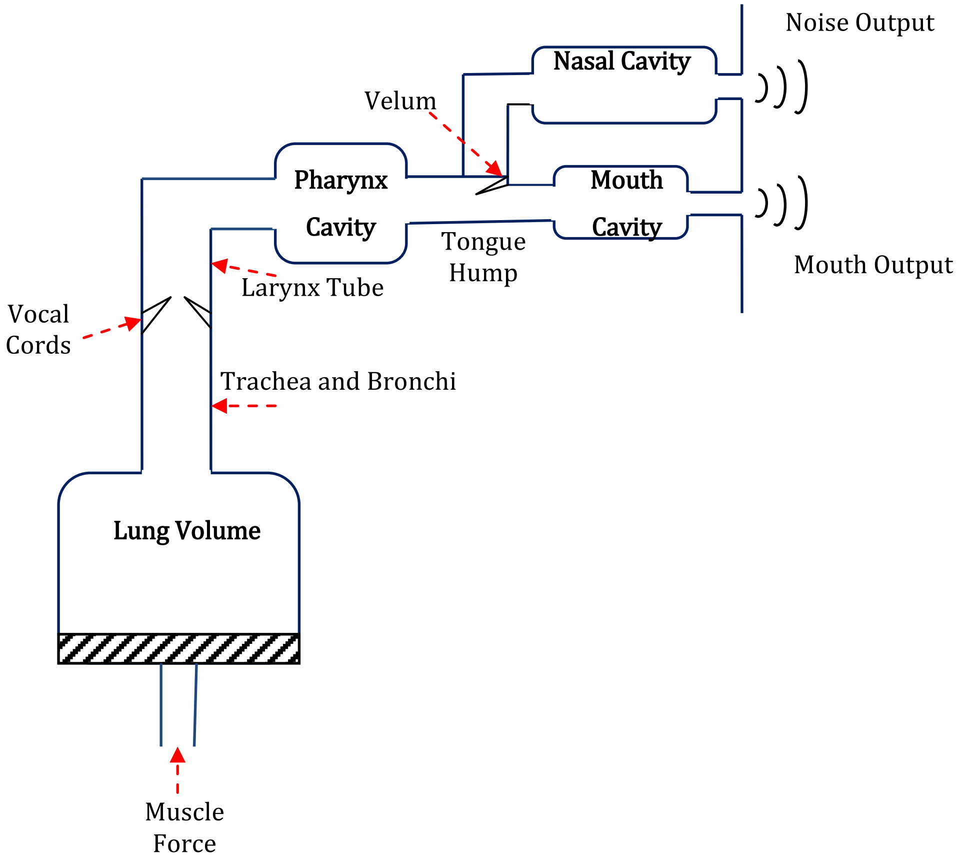


Figure 2.2: Schematic Representation of the Complete Physiological Mechanism of Speech Production [3].

2.3 Phonetic Classification of Speech

Speech is often segmented into smaller units or sounds that carry acoustical information [8]. These actual sound units that have certain acoustic and articulatory properties are called phones. Phones are realizations of phonemes, which are the basic theoretical linguistic units that comprise a word [9]. Therefore, phonemes are the smallest distinctive unit of a language, and the phones are the actual sounds of these phonemes uttered by a speaker.

Phonetic classification is the process of grouping the phonemes based on their properties pertinent to their waveforms, frequency characteristics, manner of articulation, place of articulation, type of excitation, and the stationary characteristic of the phoneme [8]. The most general classification scheme groups phonemes into two broad categories: 1) voiced speech which does not restrict the airflow through the vocal tract, and 2) unvoiced speech which restricts the airflow at some point along the vocal tract [9]. A more specific classification based on the properties mentioned above include *vowels*, *diphthongs*, *fricatives*, *affricates*, *nasals, semivowels* (*liquids* & *glides*), *stops* (*plosives*) and *whispers* (See Fig. 2.3 for details). For example, the word "tan" consists of three phonemes, each belonging to a different class of sounds. The first phoneme "t" belongs to the stop consonant class, the second phoneme "a" belongs to the vowel class, and the third phoneme "n" belongs to the nasals.

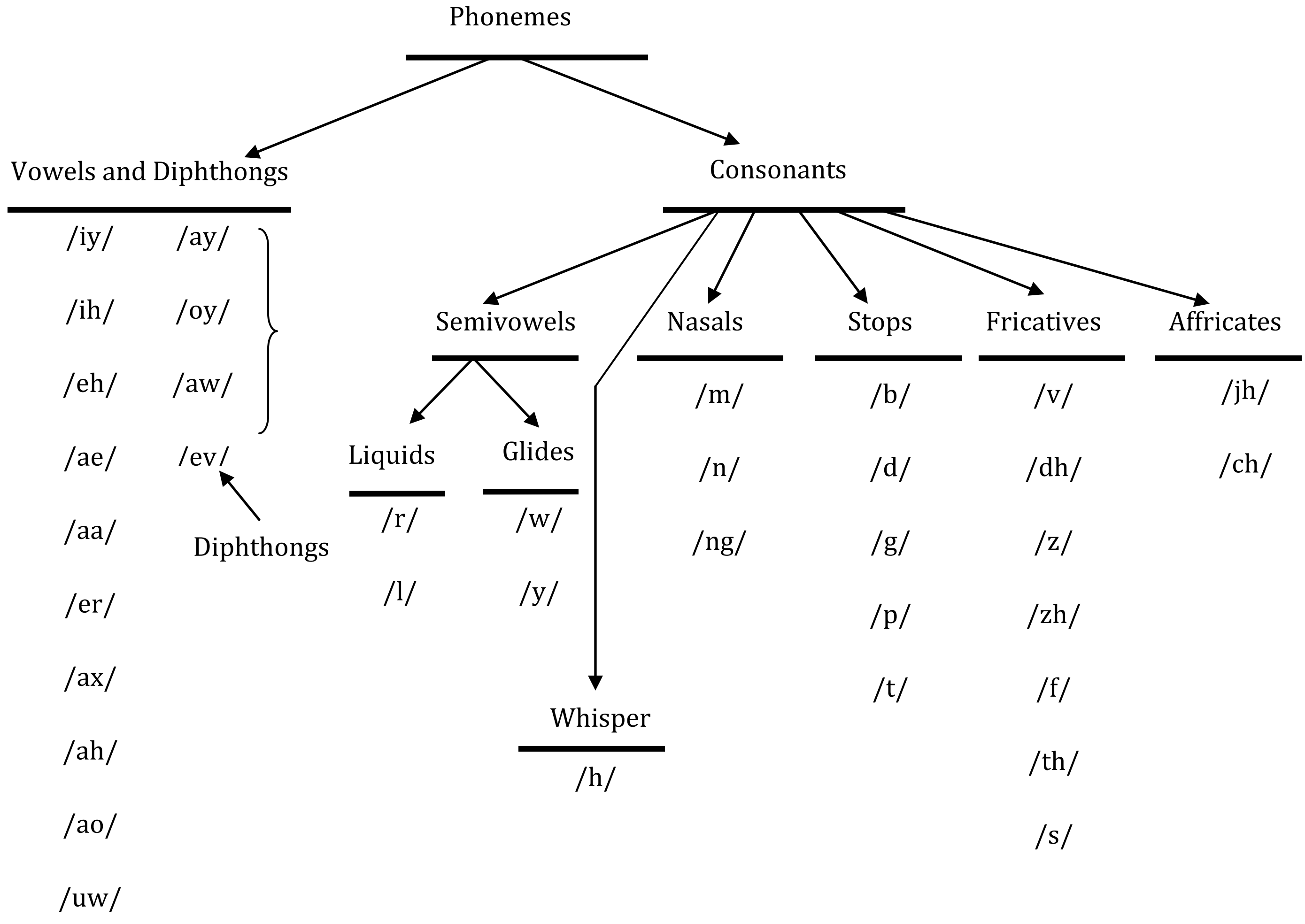


Figure2.3: Classifications of American English Phonemes into Broad Sound Classes [10].

The knowledge of the phonetic context of the vocabulary words has always been of utter importance for analyzing the implementation and performance of the endpoint detection algorithm. In general, it is a very challenging task to isolate the speech segment within the data file when the speech starts and/or ends with a weak fricative and/or a stop consonant (plosive), especially when the recording includes unwanted distortions.

2.4 Literature Review

Up until the 1990s, a limited amount of research on endpoint detection appears in the literature. The most widely referenced paper is by *Rabiner* and *Sambur* in [5]. In this paper, a fairly simple and reliable algorithm has been proposed to locate the endpoints of an utterance. The algorithm is based on two measures of speech: *Short-Time Energy* (STE) and *Zero-Crossing Rate* (ZCR). These two features of speech have been extensively used to detect the endpoints of an utterance since then. The proposed algorithm works as follows:

1. *Energy and ZCR Computations of the Framed Utterance:* For a sampling frequency of 10 kHz, a 10 ms window is chosen, and the energy is computed as the sum of the magnitude of the speech samples in this interval. The choice of a 10-ms window for computing the energy and the use of a magnitude function rather than a squared magnitude function were prescribed by the necessity to perform the computations in integer arithmetic and, thus, to increase the speed of computation. Further, the use of a magnitude (as opposed to the squared amplitude) de-emphasizes large-amplitude speech variations and produces a smoother energy function [5]. The short-time ZCR of a given frame n is defined as the number of times the successive samples of a speech sequence change sign (cross zero) per frame and given as:

$$ZCR(n) = \frac{1}{2}\sum_{i=1}^{N} |sgn[x(i+1)] - sgn[x(i)]| \tag{2. 1}$$

where N is the length of a frame, and $sgn[x(i)]$ is signum function defined as:

$$sgn[x(i)] = \begin{cases} +1, & x \geq 0 \\ -1, & x < 0 \end{cases} \quad (2.2)$$

ZCR is a reasonably good measure of the presence or absence of unvoiced speech. Moreover, using this measure as a secondary parameter in combination with short-time energy allows one to refine the initial endpoints by detecting low-energy phonemes at the beginning or end of the word [5].

2. *Threshold Setting:* Assuming that during the first 100 ms of the recording interval there is no speech present; some background silence statistics can be measured. Such statistics include the average $\overline{I_{ZC}}$ and standard deviation σ_{IZC} of the zero-crossing rate, as well as the average energy of the background silence I_{MN}.

   Subsequently, these measurements are used to set three thresholds: zero-crossing threshold I_{ZCT}, lower and upper short-time energy thresholds I_{TL} and I_{TU}. Specifically, the first threshold I_{ZCT} is set as follows:

$$I_{ZCT} = \min\left\{I_{F,}(\overline{I_{ZC}} + 2\sigma_{IZC})\right\}, \quad (2.3)$$

   where I_F is a fixed threshold, for example (25 crossings per 10 ms window length).

   The other two thresholds are set according to the following rules:

$$I_1 = 0.03(I_{MX} - I_{MN}) + I_{MN}, \quad (2.4)$$

   where I_{MX} is the peak of the energy function of the entire interval, while:

$$I_2 = 4\, I_{MN} \quad (2.5)$$

$$I_{TL} = \min\,(I_1, I_2) \quad (2.6)$$

$$I_{TU} = 5\, I_{TL} \quad (2.7)$$

3. *Searching for the initial beginning and the ending* N_1 *and* N_2 *of the utterance:* By finding the first point at which the energy exceeds I_{TL}, and then exceeds I_{TU} before falling below I_{TL}. A similar approach is used to define a preliminary estimate of the endpoint of the utterance.

4. *Refinement of endpoint estimates using zero crossing information:* This is accomplished through examining the interval from N_1 to $N_1 - 25$, (i.e., 250 ms interval preceding the initial beginning point) and counting the number of intervals where the zero crossing rate exceeds I_{ZCT}. Specifically, if this number is larger or equal to three, then the starting point is set back to the first point at which the threshold I_{ZCT} was exceeded. Otherwise, the beginning point is kept at N_1. Similarly the ending point is adjusted based on the examination of the interval N_1 to $N_1 + 25$.

The algorithm presented above serves as the basis for all energy- based endpoint detection algorithms and such approach works well enough when background noises are stationary. However, the performance of this algorithm is not very satisfactory in highly noisy environments, especially for low SNR and noises with non-stationary characteristics. It is difficult to differentiate the desired voice and unexpected background noise, such as the sound from opening or closing a door, cough sound, shaking sound from engine and so on [6].

There has been a noticeable increase of the amount of research conducted on endpoint detection after the 1990s, and many different features have been applied. In [11], *Ying et al.* have developed a new algorithm to detect the endpoints based on Teager's Energy Algortithm. Teager's Energy algorithm or simply Teager's algorithm was presented by Kaiser in [12] to compute the energy of a signal. If the samples of a signal representing the oscillatory motion of the body are given by $x_i = A\,cos(\Omega_i + \varphi)$, where A is the sample amplitude, Ω is the digital frequency in radians/sample, and φ is the initial phase in radians, then the energy of the signal is given by the following formula [12]:

$$
\begin{aligned}
E_i &= {x_i}^2 - x_{i+1}x_{i-1} \\
&= A^2 sin^2(\Omega) \approx A^2\Omega^2
\end{aligned}
\qquad (2.8)
$$

From Equation 2.8, we can see that:

- The algorithm takes into account not only the current sample, but also two adjacent samples. Thus, the instantaneous energy computed on the time-domain samples can capture dynamic changes in a signal rapidly

- Teager's energy is affected by both amplitude and frequency. Therefore, it is capable of responding rapidly in both A and Ω.

The proposed algorithm in [11] implemented energy computations on a per-frame basis instead of on a per-sample basis and called the resulting algorithm the *Frame-based Teager Energy* feature (FTE), which is computed according to the following steps:

- The power spectrum of the samples in a frame is first estimated using Discrete Fourier Transform (DFT).
- Each sample in the power spectrum is weighted with the square of its corresponding digital frequency.
- Finally, the frame energy is obtained by taking the square root of the sum of the weighted power spectrum.

In experimental part of [11], the speech data is split into overlapped frames. FTE feature is calculated for each frame as stated above. Some thresholds are determined from the first 10 frames of the recording. A search scheme applied to determine where the FTE measure first exceeds the upper threshold. Finally, endpoint locations are refined by searching along the FTE curve for the location where the FTE measure first goes below the lower threshold.

Speech endpoint detection continues to be a challenging problem particularly for speech recognition in noisy environments, and many different features have been investigated. Entropy, which is originated in the fields of coding and information theory, was first applied to the problem of endpoint detection by *Shen et al.* in [13].

Their experiments revealed that the spectral entropy of a speech segment is quite different from that of a silence one, where the short-term spectrum is more organized during speech segments than during noise, leading to relatively greater noise entropy compared to speech entropy. Based on this character, the endpoints can be properly figured out.

The spectral entropy of the n^{th} speech frame is calculated in the following manner:

- K-point Discrete Fourier Transform (DFT) is computed as follow:

$$X_n(k) = \sum_{m=0}^{M-1} x_n(m) e^{-j2\pi km/M}, \quad for\ k = 0,1, \ldots .M\text{-}1 \qquad (2.9)$$

- Spectral energy of the frequency index k in each frame is estimated as:

$$S_n(k) = |X_n(k)|^2\ , \ for\ k = 0,1, \ldots .M/2 \qquad (2.10)$$

  where the spectral energy is known to be symmetric.

- The probability density function (PDF) of the spectrum, can be estimated by normalizing the spectral energies:

$$P_n(i) = \frac{S_n(i)}{\sum_{k=0}^{M/2} S_n(k)}\ , for\ \ i = 0,1, \ldots .M/2 \qquad (2.11)$$

- To improve the discriminability of the PDF between speech and non-speech signals, two empirical constraints are applied to the PDF defined above :

$$S_n(k) = 0, \ \ if\, f < 250\ Hz\ \ or\ f > 6000Hz \qquad (2.12)$$

  This is because most of the frequency components of speech signals are covered in this region.

$$P_n(i) = 0, \quad if\ P_n(i) < \delta_2\ or\ P_n(i) > \delta_1 \qquad (2.13)$$

  Where $\delta_1$is used to eliminate the noise concentrating on some specific frequency bands, i.e. to avoid strong tones, while δ_2 is used to cancel that noise with almost constant power spectral density values over all frequencies like white noise.

- Finally, the entropy of a speech frame is defined as :

$$H_n = -\sum_{i=0}^{M/2} P_n(i)\ log_{10}[P_n(i)] \tag{2.14}$$

In the method of [12], the spectral entropy values of different frames are first evaluated and smoothed by a median filter throughout the utterance. Some thresholds are then used to detect the beginning and ending boundaries of the utterance and another set of thresholds for the refinement of the detected boundaries.

The entropy-based approach is more reliable than pure energy-based methods in some cases, particularly when the non-stationary noises are mechanical sounds. Nevertheless, experiments show that it failed under babble noise and background music. In such cases, entropy becomes very unstable. On the contrary, under those cases, energy performs well because of its additive property: energy of the sum of speech plus noise is always greater than energy of noise [6].

Consequently, many researchers have focused on the task of building more noise-robust endpoint detectors that may be operated in noisy environments. *Huang* and *Yang* in [6] have proposed a new feature that combines the two mostly wide features Energy and Entropy. This new feature has been found to be more reliable and robust, since it possesses advantages of each individual feature while compensates the drawbacks of each other.

The proposed feature is referred to as *Energy-Entropy Feature* (EEF) and is formed as follows:

- First, both energy (sum of squared samples) and entropy (as described in [13]) are computed in parallel for each frame.
- Then their reference values are subtracted to shift their base lines. This is achieved by subtracting the average amount of the first 10 frames accordingly. Finally, the adjusted values are multiplied to get the proposed feature

$$M_n = (E_n - C_E).(H_n - C_H) \tag{2. 15}$$

$$EEF_n = \sqrt{1 + |M_n|} \qquad (2.16)$$

where C_E and C_H denote the average energy and average entropy of the first 10 frames, respectively. After computing the EEF for each frame, simple decision logic is then used with a pair of thresholds to determine the final endpoints. The experimental results show that this approach has a higher accuracy than energy- based algorithms [6].

In [14], *Junqua* et al. have proposed the *Time-Frequency* (TF) parameter to detect speech, which assumes that frequency information in the frequency ranges 250–3500Hz is less contaminated by noise. The TF parameter is composed of both frequency energy in the fixed frequency bands and time energy.

Based on the TF parameter, the algorithm was proposed to get more precise word boundary detector in noisy environment. This algorithm can be described as follow:

- First, the energy in the frequency band (250-3500Hz) is computed, normalized and smoothed by a median average algorithm.
- The logarithm of the non-bandlimited root mean square energy is then computed, normalized, and smoothed.
- The final parameter used (TF) is the result obtained after smoothing the sum of the two energy curves.
- Then, a noise adaptive threshold is computed from the first few frames of the speech signal to determine the beginning of the first vowel and the end of the last vowel.
- Finally, a refinement procedure is applied.

Although this algorithm outperforms several commonly used alternatives for word boundary detection in the presence of noise, it requires one to determine thresholds empirically (using somewhat ambiguous rules), which is rather inconvenient from the practical point of view [7].

Wu and *Lin* in [7] have modified the TF parameter approach by proposing *Adaptive Time Frequency* (ATF) parameter for extracting both time and frequency features of noisy speech signals. The ATF parameter is performed using *Adaptive Band Selection* (ABS) as a noise cancellation method. The ATF parameter can extract useful frequency information by adaptively choosing proper frequency bands of the Mel-scale frequency bank. The author proposed a new word boundary detection algorithm by using a *Self-Constructing Neural Fuzzy Inference Network* (SONFIN) for identifying islands of word signals in noisy environment. Due to the self-learning ability of SONFIN, the proposed algorithm avoids the need of empirically determining the thresholds. The proposed approach has been shown to be able to reduce the recognition error rate to about 10% as compared with TF-based algorithm [7].

Although ATF-based algorithm outperforms many algorithms used for endpoint detection, it is found that the selection of useful bands depends on the information of the whole recording. Additionally, ATF parameter is based on energy, which is less reliable in the presence of non-stationary noise. *Wu* and *Wang* in [15] have found that the frequency energies of various types of noise are concentrated in different frequency bands and the inherent characteristic of banded nature is robust to noise. As a result, they have proposed a new feature called *Band-Partitioning Spectral Entropy* (BSE). To select useful bands effectively and accurately, a *Refined Adaptive Band Selection* (RABS) method was also proposed, which is extended from the *Adaptive Band Selection* ABS method presented in [7]. Finally, the RABS method incorporated the BSE parameter to form a new *Adaptive Band-Partitioning Spectral Entropy* (ABSE) feature to detect endpoints effectively under conditions of low SNR.

The experimental results reported in [15] reveal that the ABSE-based algorithm performs reliably in the presence of four types of noise (vehicle, multi-talker babble, factory, and white noise) at various SNR levels. The algorithm has also been shown to perform successfully in real cars with musical background noise. The entropy-based parameter is related only to the variation of spectral energy but not to the amount of spectral energy, so the ABSE-based algorithm outperforms the energy-based algorithm, especially in changing level of noise [15].

In [16], *Yingle et al.* have proposed applying a time-frequency speech enhancement stage prior to spectral entropy endpoint detection algorithm introduced in [13]. In this paper, the noisy speech is enhanced using Spectral Subtraction method in the frequency domain to remove the additive noise. In order to remove the residual noise produced by the spectral subtraction, a weighting function in the time domain is constructed by the original speech short-time energy and zero-crossing rate. Finally, spectral entropy based method is used to locate the endpoints. Experimental results showed that the proposed algorithm outperforms the entropy based without using the enhancement stage prior to endpoint detection, especially for low SNRs [16].

Change-point detection method has been also applied to the problem of endpoint detection as done in [17] by *Lipeika* and *Lipeikiene*. The proposed method was based on the assumption that there are two change-points in the signal, viz. the beginning and the end of a spoken word. They used fixed length segments at the beginning and the end of the signal to estimate initial background noise parameters, while the complementary portion of the signal was used to estimate initial energy parameters of the spoken word. Subsequently, the likelihood maximization based on dynamic programming was used to estimate endpoints using the initial bounds. The spoken word and background noise parameters were then re-estimated according to resulting endpoints, and this procedure is repeated until endpoint estimates stop changing. The main advantage of this approach is that it avoids the need of thresholds and heuristic decision rules due to the incorporation of dynamic programming.

Statistical modelling for endpoint detection has attracted considerable attention, and many researchers focused on finding suitable model to simulate the empirical distribution of the speech. In [18], *Sohn* et al. have proposed statistical model-based voice activity detection (VAD) as a robust decision mechanism. This method constructs a statistical model by using an ergodic state transition model with speech and non-speech states. Then it calculates the likelihood ratio of a speech state to a non-speech state based on Hidden Markov Model (HMM) and makes use of Likelihood Ratio Test (LRT) to discriminate between speech and non-speech frames.

The statistical VAD proposed in [18] uses the conventional a priori and a posteriori SNR-based approach to calculate likelihood for each state, which is not directly calculated by using any kind of probability density function (PDF). Since the likelihood calculation with PDFs is more flexible and applicable, *Fujimoto et al.* in [19] have proposed Gaussian Mixture Models (GMMs) of noise (noise + silence) and noisy speech (noise + speech), and calculated the likelihood of speech and non-speech states directly.

Furthermore, Sohn's statistical VAD [18] was derived under the assumption that noise has stationary characteristics. Unfortunately, this assumption is impractical; since most of the noise characteristics observed in real environments are non-stationary. To overcome this problem, the authors in [19] introduced the using of a parallel non-linear Kalman filter. In addition, backward techniques (such as parallel Kalman smoother and backward probability estimation) have been used for noise estimation and likelihood calculation for speech and non-speech discrimination. The evaluation results showed that the proposed method significantly improves VAD accuracy compared with conventional methods [19].

The statistical models could detect the voice activity precisely, but they are not efficient in practice. Thus, *Wu* and *Zhang* in [20] have presented a new VAD that combines statistical models and empirical rule-based energy detection algorithm. In this study, the energy detection sub-algorithm is first used to detect the possible endpoints. However, these endpoints are not accurate enough in the case of noisy speech. Accordingly, the authors proposed a new *Gaussian Mixture Model-based Multiple Observation Log Likelihood Ratio* (GMM- based MO-LLR) algorithm to align the endpoints with their optimal positions. The experimental results showed that the proposed algorithm could achieve a better performance than some commonly used VADs. It has also been demonstrated that the proposed VAD is more efficient and robust in different noisy environments [20].

In [21], *Zhang* and *Hu* have proposed a new endpoint detection algorithm based on Mel-Frequency Cepstral Coefficients (MFCC) and spectral entropy. The algorithm uses the 12-order MFCC parameters and spectral entropy proposed in [13] as a feature vector. As well, it employs a trained *Back Propagation Neural Network* (BP NN) as a classifier to distinguish the speech and non-speech segments from audio signals, so as to avoid the need to set thresholds. Also, there is no need to assume that the first few frames of signals are noisy signals. Experimental results in [21] indicated that the proposed method is more reliable and efficient than the traditional ones based on short-term energy at low SNR [21].

Some other methods were also proposed for the Voice activity detection using discrete wavelet transform as in [22] by *Aghajani et al.* In this approach, the energy of each sub-band is determined from the wavelet coefficients, resulting in a feature vector. Finally, the Euclidian distance between feature vector of the frame and the noise feature vector is calculated and compared to a predetermined threshold value. Experimental results demonstrated advantage of this algorithm over different VAD methods [22].

Several works have been seeking to solve the problem of endpoint detection in noisy environments. *Ghaemmaghami* et al. in [23] has been the first to develop a method that utilizes gradient based edge detection algorithms, original from image processing field, to perform boundary detection for continuous speech in noisy environments. Gradient based image processing edge detectors localise edges within an image based on rapid gradient change at edge boundaries. This algorithm estimates a speech utterance region through observation of the time-domain plot of the associated noisy speech signal. Hence, the speech signal is converted into a matrix and treated as an image to obtain an accurate estimation of the speech utterance regions or boundaries (edges) within the noisy signal. It is shown that the proposed method outperforms some VAD algorithms over a range of SNR levels, noise types and signal lengths. However, the method is not yet suitable for real-time applications and assumes a single utterance region per input vector. This is due to the decision smoothing technique employed in the proposed method.

Chapter 3

Speech Endpoint Detection: An Image Segmentation Approach

3.1 Introduction

Similar issues to speech endpoint detection have also been studied in other research areas, such as edge detection in image processing [41], and change-point detection in theoretical statistics [42]. It can be seen that if an extracted feature from an audio signal containing speech is plotted against time, the mean value of this feature's energy over both the speech and non-speech portions can be clearly distinguished. This observation raised the question of whether it would be possible to modify such an image segmentation algorithm that is based on energy minimization to be applied in the field of speech detection. Conducting research in this direction has led to the development of the proposed algorithm which consists of four main stages, as shown in Fig. 3.1. In the first stage, the speech signal is enhanced using frequency domain multiband spectral subtraction. The second stage includes a number of preprocessing steps which are applied to the speech signal before any speech- specific information is extracted. The third stage of the algorithm is the extraction of three different features of the speech signal which are required to convert the speech waveform to a parametric representation at a lower information rate for further analysis. Finally, the extracted features are processed by a novel detection algorithm in order to solve the problem of endpoint detection.

More details about the algorithm will be discussed thoroughly in the following sections.

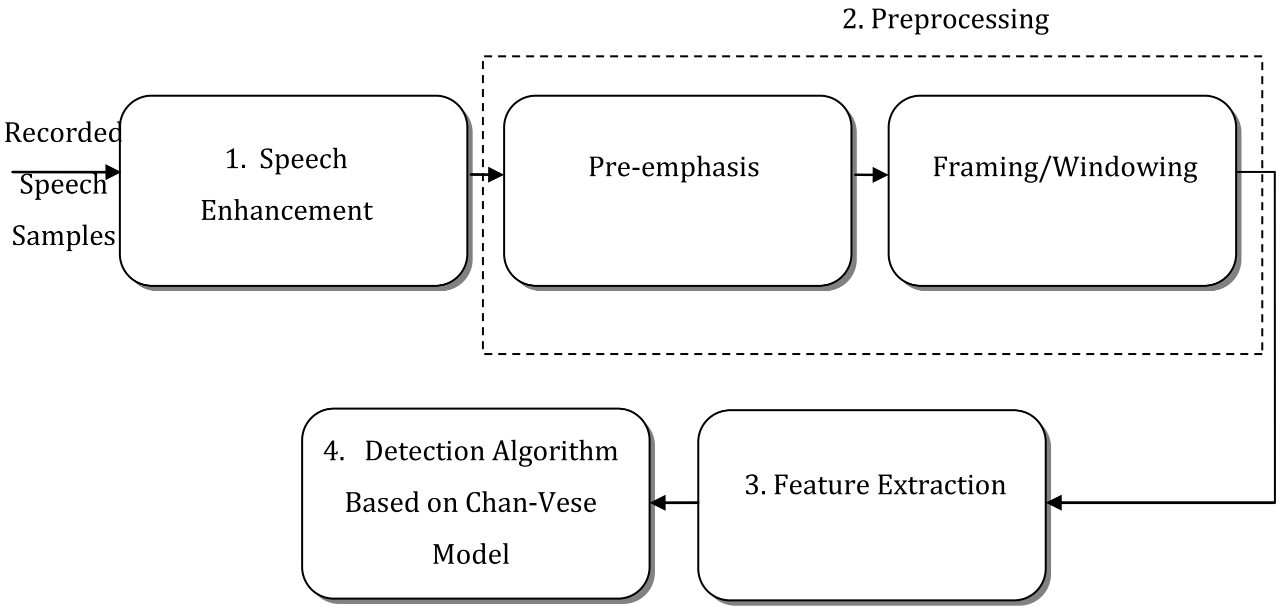


Figure 3.1: Block Diagram of the Proposed Algorithm.

3.2 Speech Enhancement

To increase the accuracy of the proposed method, the measurement noise is first rejected by means of a speech enhancement method. In this thesis, we use a *Multi-Band Spectral Subtraction* (MBSS) approach, which is a variation of the basic spectral subtraction technique [10].

While the conventional spectral subtraction techniques reduce the noise level (thereby improving the speech quality), it may also introduce an undesirable distortion called *musical noise*. This distortion is caused due to the inaccuracies in the short-time noise spectrum estimate resulting in large spectral variations in the enhanced spectrum.

The motivation behind using multiband spectral subtraction stems from the fact that, in general, noise is unlikely to affect the speech signal uniformly over the whole frequency domain. In other words, some frequencies will be affected more adversely than the others, depending on the spectral characteristics of the noise.

In the multiband approach, the speech spectrum is divided into *N* non-overlapping bands, followed by spectral subtraction performed independently in each band. The process of splitting the speech signal into different bands can be performed in the frequency domain by using appropriate windowing.

The estimated speech spectrum in the ith band can be obtained according to:

$$\left|\hat{X}_i(\omega_k)\right|^2 = |\bar{Y}_i(\omega_k)|^2 - \alpha_i.\delta_i.\left|\hat{D}_i(\omega_k)\right|^2 \qquad b_i \le \omega_k \le e_i\,, \tag{3.1}$$

where $\omega_k = 2\pi k/N$ $(k = 0,1,\dots,N-1)$ are the discrete frequencies, $\left|\hat{D}_i(\omega_k)\right|^2$is the estimated noise power spectrum (obtained and updated during speech-absent segments), b_i and e_i are the beginning and ending frequency bins of the i th frequency band, $\alpha_i$is the over-subtraction factor of the i th band, and δ_i is a band-subtraction factor that can be individually set for each frequency band to customize the noise removal properties. $\bar{Y}_i(\omega_k)$ is the i th frequency band estimation of the smoothed and averaged noisy speech spectrum as defined in the following equation:

$$\left|\bar{Y}_j(\omega_k)\right| = \sum_{i=-M}^{M} W_i\left|Y_{j-i}(\omega_k)\right|\,, \tag{3.2}$$

where $|Y_i(\omega_k)|$ is the noisy magnitude spectrum, and $W_j(0 < W < 1)$ are the weights assigned to each frame. Here, the averaging is performed over M preceding and succeeding frames of speech.

The band-specific over-subtraction factor α_i in Equation 3.1 is a function of the segmental SNR_i of the ith frequency band, and is given by [10]:

$$\alpha_i = \left\{\begin{matrix} 4.75 & SNR_i < -5 \\ 4 - \dfrac{3}{20}(SNR_i) & -5 \le SNR_i \le 20 \\ 1 & SNR_i > 20 \end{matrix}\right\} \tag{3.3}$$

where the band SNR_i is defined as:

$$SNR_i(dB) = 10log_{10}\left(\frac{\sum_{\omega_k=b_i}^{e_i}|\bar{Y}_i(\omega_k)|^2}{\sum_{\omega_k=b_i}^{e_i}\left|\hat{D}_i(\omega_k)\right|^2}\right) \tag{3.4}$$

While using the over-subtraction factor α_i provides a certain degree of control over the noise subtraction in each band, the use of multiple frequency bands as well as the δ_i weights provide an additional degree of control within each band. The values of δ_i in Equation 3.1 are empirically determined and set to [10]:

$$\delta_i = \begin{cases} 1 & f_i \leq 1kHz \\ 2.5 & 1kHz < f_i \leq \frac{F_s}{2} - 2kHz \\ 1.5 & f_i > \frac{F_s}{2} - 2kHz \end{cases} \qquad (3.5)$$

where f_i is the upper frequency in the ith band, and F_s is the sampling frequency in Hz.

The negative values resulting from the subtraction in Equation 3.1 are floored to the noisy spectrum as:

$$\left|\hat{X}_i(\omega_k)\right|^2 = \begin{cases} \left|\hat{X}_i(\omega_k)\right|^2 \; if \; \left|\hat{X}_i(\omega_k)\right|^2 > \beta|\bar{Y}_i(\omega_k)|^2 \\ \beta|\bar{Y}_i(\omega_k)|^2 \qquad\qquad else \end{cases}, \qquad (3.6)$$

where the spectral floor parameter β is set to 0.002.

The block diagram of the multiband method is shown in Fig. 3.2. In the first stage, the signal is windowed and the magnitude spectrum is estimated using DFT. Subsequently, the noisy speech spectrum is preprocessed by Equation 3.2 to produce a smoothed estimate of the spectrum. Next, the noise and speech spectra are split into N frequency bands and the over-subtraction of each band α_i is calculated. Then, the individual frequency bands of the estimated noise spectrum are subtracted from the corresponding bands of the noisy speech spectrum. Finally, the modified frequency bands are recombined and the enhanced speech signal is obtained by taking IDFT of the enhanced spectrum using the noisy speech phase.

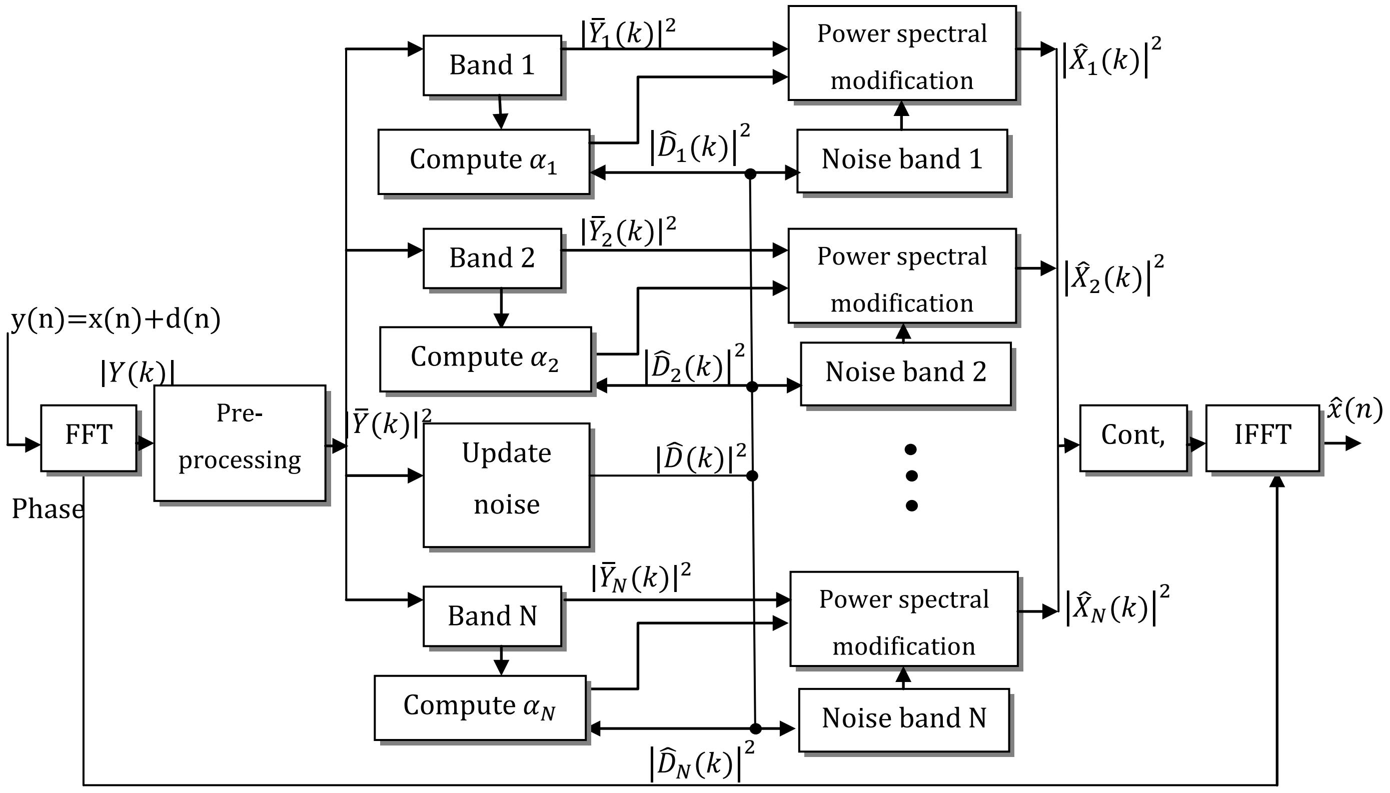


Figure 3.2: Block Diagram of the Multiband Spectral Subtraction Algorithm [10].

3.3 Signal Preprocessing

The second module of the proposed algorithm consists of the preprocessing of input speech data. In this module, the input speech data are subjected to signal processing to enhance the *feature extraction* performed in the next stage.

The second module is composed of two processing stages, viz. *Pre-emphasis, Framing and Windowing,* which are discussed in details in the following subsections.

3.3.1 Pre-emphasis

In most of speech analysis applications, speech waveforms are usually pre-emphasized prior to extracting speech features. The pre-emphasis is achieved by applying a first order digital filter that increases the relative energy of the high-frequency of the speech. This high-pass (HP) filter can be defined by the following transfer function:

$$H_{pre}(z) = 1 - \mu z^{-1}, \tag{3.7}$$

where μ is a filter parameter, which determines its cut-off frequency. Typically, μ takes a value in the range $0.9 \leq \mu \leq 1$. The frequency response of the pre-emphasis filter for μ=0.94 is shown in Fig. 3.3.

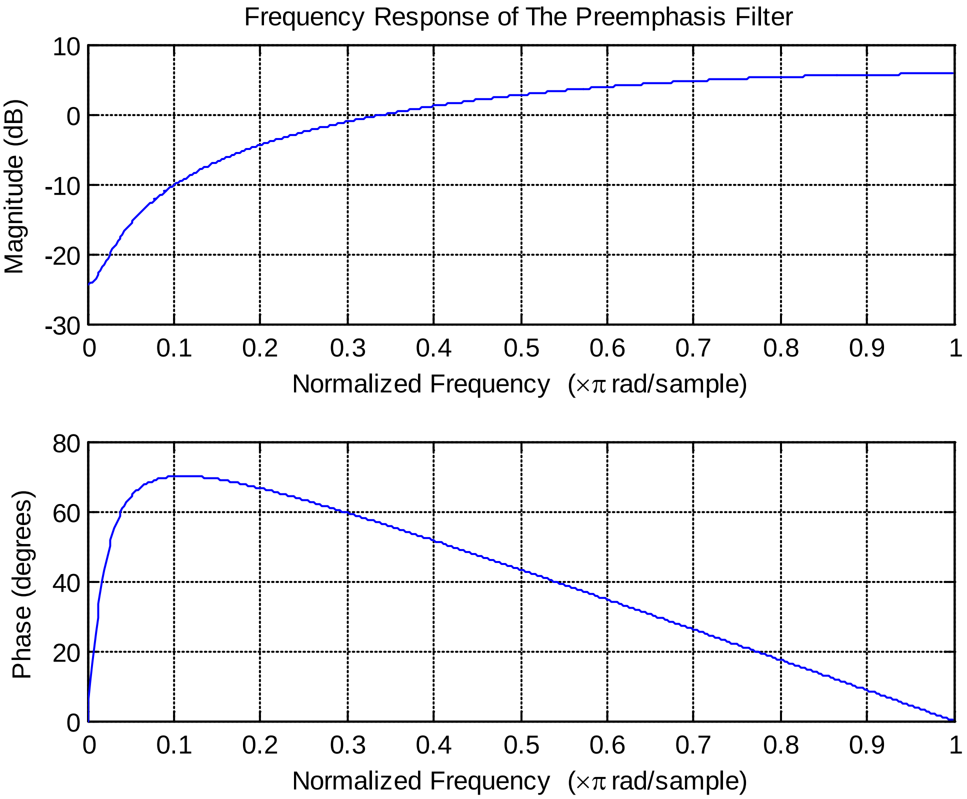


Figure 3.3: Pre-emphasis Filter Frequency Response.

The pre-emphasis filter actually models the lip radiation characteristics, and introduces a zero near ω = 0, and a 6-dB per octave shift on the speech spectrum [8]. There are several reasons for employing such a pre-emphasis filter. First, using the filter tends to cancel the glottal or lip radiation effects on speech production so that one can achieve more accurate results when representing the whole speech production procedure with the vocal tract model filter [8]. Another reason for using the pre-emphasis stage is to prevent numerical instability. If a speech signal is dominated by low frequencies, its autocorrelation matrix might turn out to be singular, and as a result, its inversion will cause numerical instability.

Finally, the pre-emphasis filter can also help to boost the signal spectrum (approximately 20 dB per decade). This can have many applications in case of voiced phonemes. It can also be noted that such pre-emphasis filters tend to raise frequencies above 5 kHz, a region in which the auditory system becomes increasingly less sensitive. Moreover, the frequencies above 5 kHz are naturally attenuated by the speech production system [24].

3.3.2 Framing and Windowing

In speech processing, before extracting the features, it is common to segment the speech waveform into finite-length frames followed by windowing. In this module, the input signal is divided into overlapped frames of length *M.* Each of these frames is then multiplied by a window function. Note that, in combination with overlapping short-term frames, successive windowing amounts to applying a sliding window to the original speech signal.

Fig. 3.4 illustrates the concept of signal framing, where each frame shares the first part with the previous frame and the last part with the next frame.

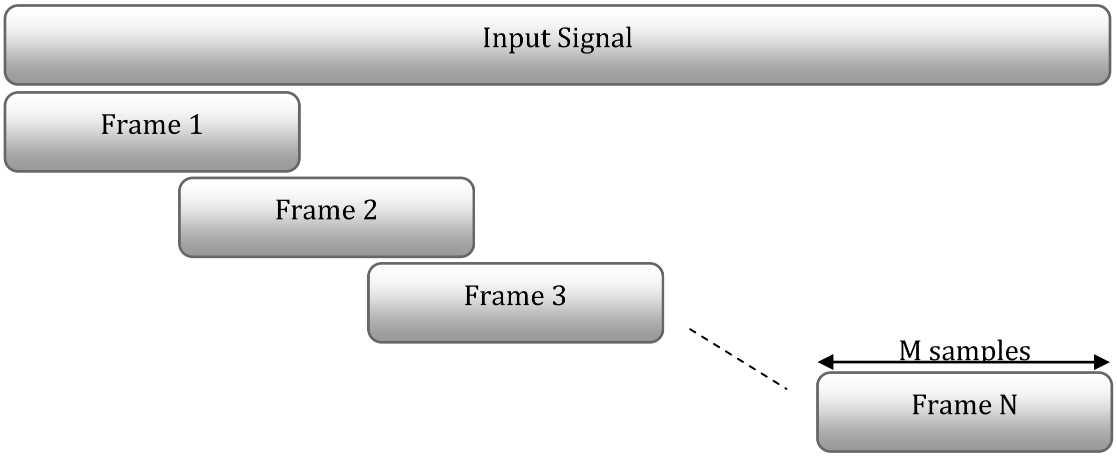


Figure 3.4: Framing and Overlapping.

In general, speech signals are not stationary (i.e. their statistical characteristics may vary in time). The lack of stationarity is caused by the changes of the vocal tract during speech production. However, when restricted to a short-time interval, the speech signal can be considered to be quasi-stationary, because of the fact that the glottal system can not change immediately. Generally speaking, the use of short frame duration and overlapping frames is chosen to capture the rapid dynamics of the spectrum. Therefore, the choice of the frame and overlap lengths are very important.

In practical systems, frame duration typically ranges between 10 msec and 30 msec . A specific value in this range is chosen to optimally balance between the rate of change of spectrum and system complexity [24].

The overlapping of speech frames is used in order to increase the redundancy of the input signal, to provide more speech data to the feature extraction algorithms. Moreover, we can capture the changes in the vocal tract more accurately. The extent of the overlap depends on a particular signal/system, with a common choice being 50%.

In this thesis, a Hamming window function has been applied to the frames to minimize the discontinuity of the signal at the beginning and the end of resulting frames. The Hamming window function (as shown in Fig. 3.5) is given by the following equation:

$$w[m] = 0.54 - 0.46 \cos\left(\frac{2\pi m}{M-1}\right), for\ \ m = 0,1, \ldots..M-1\ , \tag{3.8}$$

where m represents the sample number and M is the total number of samples in a frame.

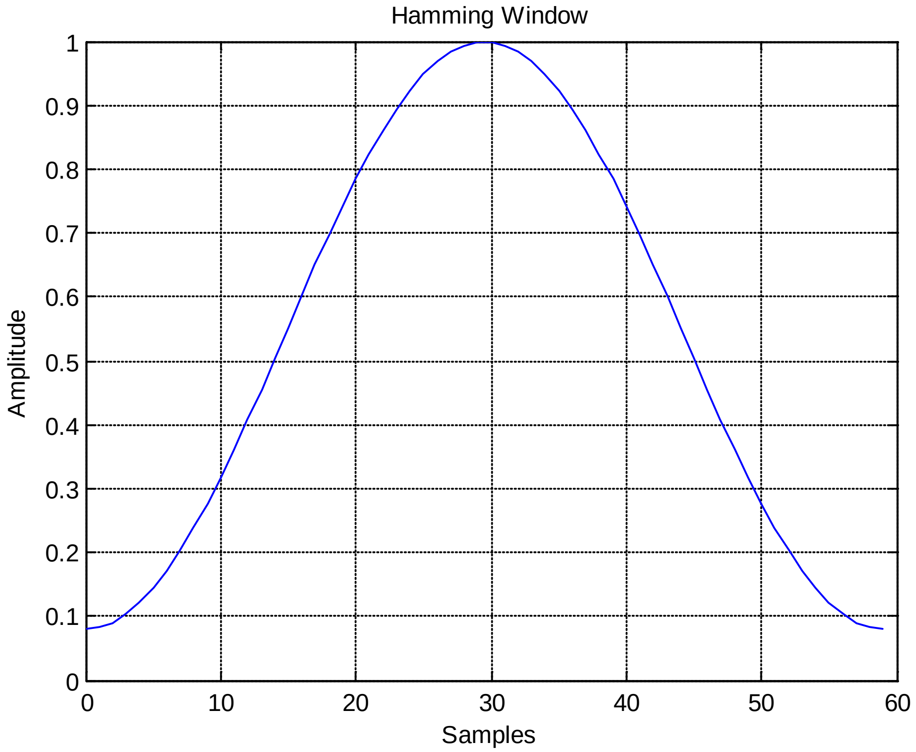


Figure 3.5: Hamming Window Function for M=60.

3.4 Feature Extraction

After performing all the necessary preprocessing to the input speech, the next step will be the *Feature Extraction* module. The basic role of this module is to use the input speech samples to calculate certain parameters (features) that will be used in the detection stage.

The key to having a high accuracy rate is selecting the appropriate features to optimize the performance of the speech detection system. These features should be able to distinguish between speech and background silence and they need to be robust to ambient noise.

Different features of the speech signal contain various information about the speech waveform. Each feature requires a different methodology and a different level of computation complexity to extract it. In the following subsections, we will examine the extraction methodology of three features (Log- Energy, PLP, and MFCC), which will be used in our algorithm.

3.4.1 Log-energy

The short-time energy has been used extensively as a speech feature in many endpoint detection algorithms since 1970's, because of its simplicity and ease of implementation. The short-time energy is also a natural way of representing the amplitude changes in speech signals. Some segments of a speech signal, such as unvoiced segments, tend to have much lower amplitude than the voiced segments. As a result, the energy of such unvoiced segments should be lower than their voiced counterparts. Therefore, the energy measure can be used as a feature to discriminate between voiced and unvoiced segments (subject to appropriate thresholding). In particular, the short-time energy measure can also be used to discriminate between speech and silence segments in environments with very high signal-to-noise ratios (30 dB or higher), where the lowest energy segments of the speech signal will exceed the energy of the silence segments [5].

There are various ways to calculate the energy of a speech signal. The most used ones are:

- *Squared Energy :*

$$E_k = \sum_{i=1}^{M} |x_k(i)|^2 \tag{3.9}$$

- *Root Mean Square Energy (RMSE)* :

$$E_k = \sqrt{\frac{1}{M}\sum_{i=1}^{M} |x_k(i)|^2} \tag{3. 10}$$

- *Absolute Magnitude Energy :*

$$E_k = \sum_{i=1}^{M} |x_k(i)| \tag{3. 11}$$

- *Log-Energy* :

$$E_k = \textstyle\sum_{i=1}^{M} log_{10}|x_k(i)|^2 \, , \tag{3.12}$$

where M denotes the width of the window used to segment the speech into N numbers of frames, $x_k(i)$ represents the i^{th} windowed speech sample in frame k, and E_k is the energy of frame k.

The *squared energy* measure suppresses low-frequency noise completely, and it is more stable than other measures. However, low energy segments, such as the weak fricatives and stop consonants, are likely to be deemphasized with the background noise. Thus, the squared energy provides very conservative edge point estimates and can be used to detect the voiced (mainly, vowel) portions of the words [25].

The *RMS energy* resembles the squared energy in the sense that it is a scaled version of the squared energy parameter. The square root operator, however, emphasizes low energy segments while deemphasizing higher energy ones, which also makes it behave similarly to the absolute magnitude energy. This characteristic in turn reduces the relatively large energy difference between voiced and unvoiced segments composing an utterance.

The *absolute magnitude energy* represents a sum of the magnitudes of signal values in a given frame; hence, the weak unvoiced segments of the utterance are not deemphasized, and as a result, this quantity is capable of detecting information about the speech frame. However, some detection schemes may become unstable in strong noise cases since the background noise is not suppressed at all [25].

In the proposed algorithm, *Log-Energy* feature has been used, since the logarithm function results in a non-linear compression of signal amplitude and it is capable to detect the relatively weak amplitudes of the signal [25]. The normalized log-energy feature, corresponding to the word “Hot”, is depicted in Fig. 3.6.

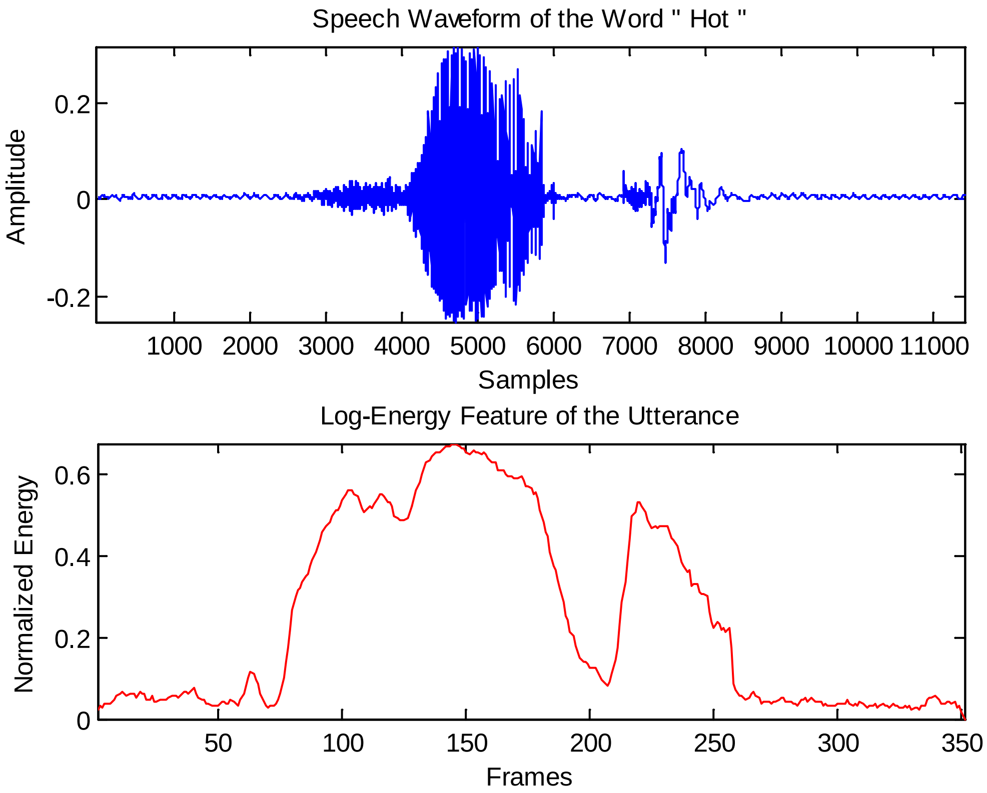


Figure 3.6: Waveform of the Word ”Hot”(Top Plot), and Corresponding Energy Curve (Bottom Plot).

3.4.2 Perceptual Linear Predictive Coefficients

The Perceptual Linear Prediction (PLP) is a relatively new technique for the analysis of speech proposed by Hermansky in [26]. As opposed to the conventional linear predictive analysis (LP), the PLP analysis is known to be more consistent with the human auditory system. This technique uses three concepts from the psychophysics of hearing to derive the auditory spectrum estimation, viz.: (i) critical- band spectral resolution, (ii) equal-loudness curve and (iii) intensity-loudness power law. The auditory spectrum is then approximated by an autoregressive (AR) model. Fig. 3.7 shows a block diagram of PLP coefficients calculation of a speech segment.

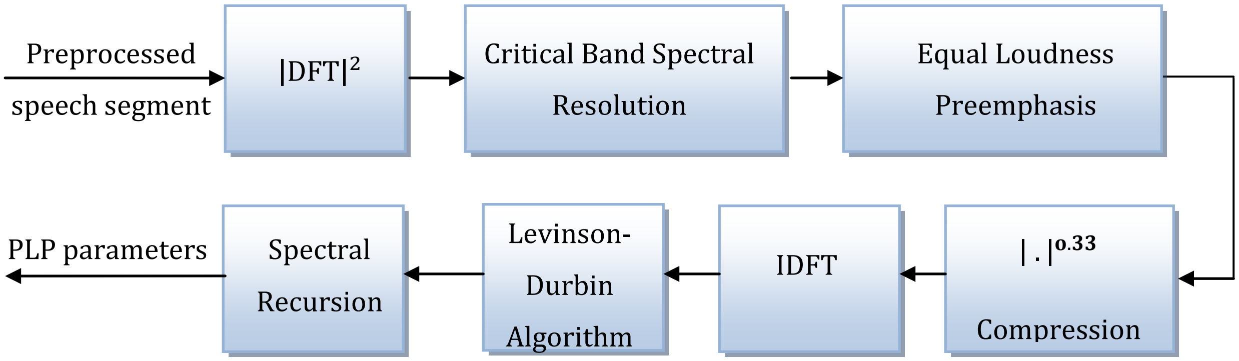

Preprocessed speech segment
|DFT|²
Critical Band Spectral Resolution
Equal Loudness Preemphasis
| . |^0.33 Compression
IDFT
Levinson-Durbin Algorithm
Spectral Recursion
PLP parameters

Figure 3.7: Block Diagram of Perceptual Linear Predictive (PLP) Analysis of Speech.

Below, we provide detailed explanation of each block of this technique:

I. Spectral Analysis

The preprocessed speech segment is first transformed into the frequency domain by taking the Discrete Fourier Transform (DFT). Then, the short-term power spectrum $P(\omega)$ is obtained by squaring the absolute value of the DFT.

II. Critical-band Spectral Resolution

The power spectrum $P(\omega)$ of the signal is first warped along its frequency axis ω into the Bark frequency Ω by the following equation [26]:

$$\Omega(\omega) = 6\,ln\left(\frac{\omega}{1200\pi} + \left(\left(\frac{\omega}{1200\pi}\right)^2 + 1\right)^{0.5}\right) \quad , \qquad (3.13)$$

where ω is the angular frequency in rad/sec. Fig. 3.8 shows the mapping of the frequency to the Bark scale.

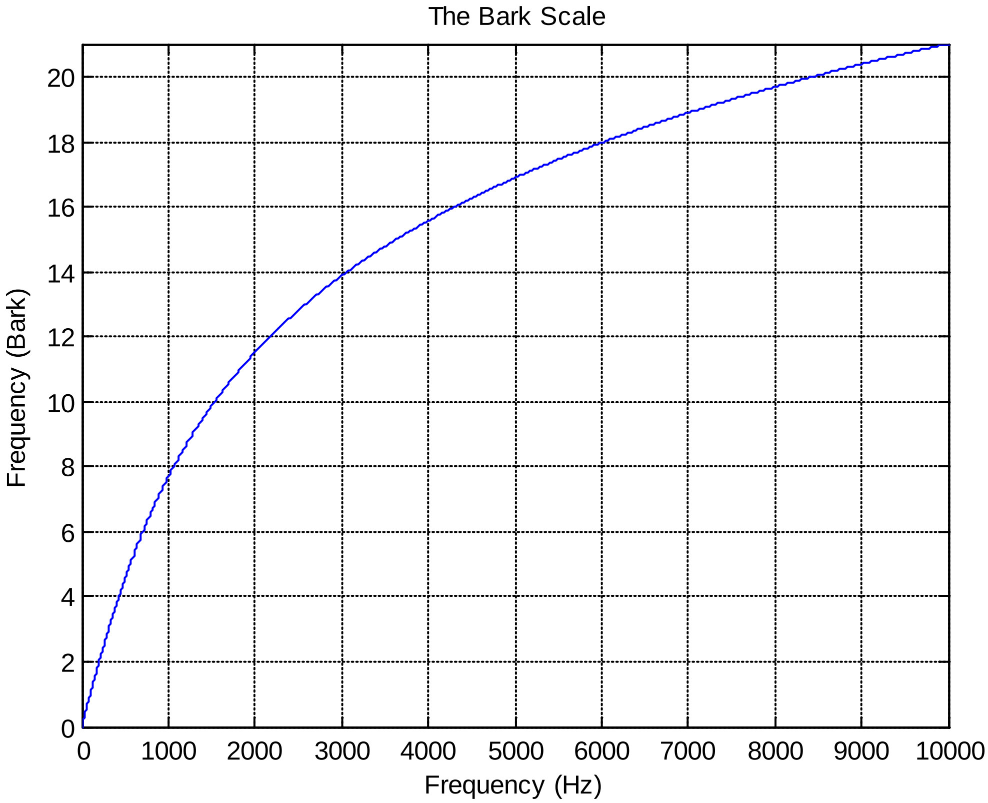


Figure 3.8: The Bark Scale.

Subsequently, the resulting warped power spectrum is convolved with the critical-band masking curve $\psi(\Omega)$ defined by:

$$\psi(\Omega) = \begin{Bmatrix} 0, & for\ \Omega < -1.3 \\ 10^{2.5(\Omega+0.5)}, & for - 1.3 \leq \Omega \leq -0.5 \\ 1, & for - 0.5 < \Omega < 0.5 \\ 10^{-(\Omega-0.5)}, & for\ 0.5 \leq \Omega \leq 2.5 \\ 0, & for\ \Omega > 2.5 \end{Bmatrix} \qquad (3.14)$$

In practice, trapezoidal shaped filters are commonly applied to the power spectrum at bark intervals, where the Bark axis is derived from the frequency axis using the warping function given by Equation 3.13. The discrete convolution of $\psi(\Omega)$ with (the even symmetric and periodic function) $P(\Omega)$ yields samples of the critical-band power spectrum:

$$\theta(\Omega_i) = \sum_{\Omega=-1.3}^{\Omega=2.5} P(\Omega_i - \Omega)\,\psi(\Omega) \qquad (3.15)$$

III. Equal-loudness Pre-emphasis

The resulting $\theta[\Omega(\omega)]$ is pre-emphasized by the simulated equal loudness curve:

$$\Xi[\Omega(\omega)] = E(\omega)\theta[\Omega(\omega)] \tag{3.16}$$

where the function $E(\omega)$ approximates to the non-equal sensitivity of human hearing at different frequencies, thereby simulating the sensitivity of hearing at about the 4O-dB level. This function is given by [26]:

$$E(\omega) = \frac{(\omega^2 + 56.8 * 10^6)\omega^4}{(\omega^2 + 6.3 * 10^6)^2(\omega^2 + 0.38 * 10^9)} \quad , \tag{3.17}$$

which is a transfer function of a filter with asymptotes of 12 dB/oct between 0 and 400 Hz, 0 dB/oct between 400 and 1200 Hz, 6 dB/oct between 1200 and 3100 Hz, and 0 dB/oct between 3100 Hz and the Nyquist frequency. This function $E(\omega)$ is known to provide a close approximation in the range up to 5000 Hz [26].

IV. Intensity-loudness Power Law

The last operation prior to the AR modeling is the cubic-root amplitude compression, which is performed according to:

$$\Phi(\Omega) = \Xi(\Omega)^{0.33} \tag{3.18}$$

This operation is an approximation to the power law of human hearing and it simulates the nonlinear relation between the intensity of sound and its perceived loudness [26].

V. Autoregressive modeling

After all the previous operations, all signal components are perceptually equally weighted and a regular Linear prediction (LP) model can be estimated. In this step, the AR modelling is applied to the real part of the IDFT of $\Phi(\Omega)$ by using the *Levinson-Durbin* algorithm. Note that the resulting AR parameters could be further transformed into a different set of parameters, such as spectral coefficients of the AR model [26].

In this thesis, the order of the PLP model has been set to 5; since it has been found that the 5th-order PLP analysis is consistent with the sensitivity of human hearing to changes in several important speech parameters [26]. As the representing feature, we take the mean value of the estimated coefficients. Fig. 3.9 shows the normalized averaged PLP feature for the recording of the word “Hot”.

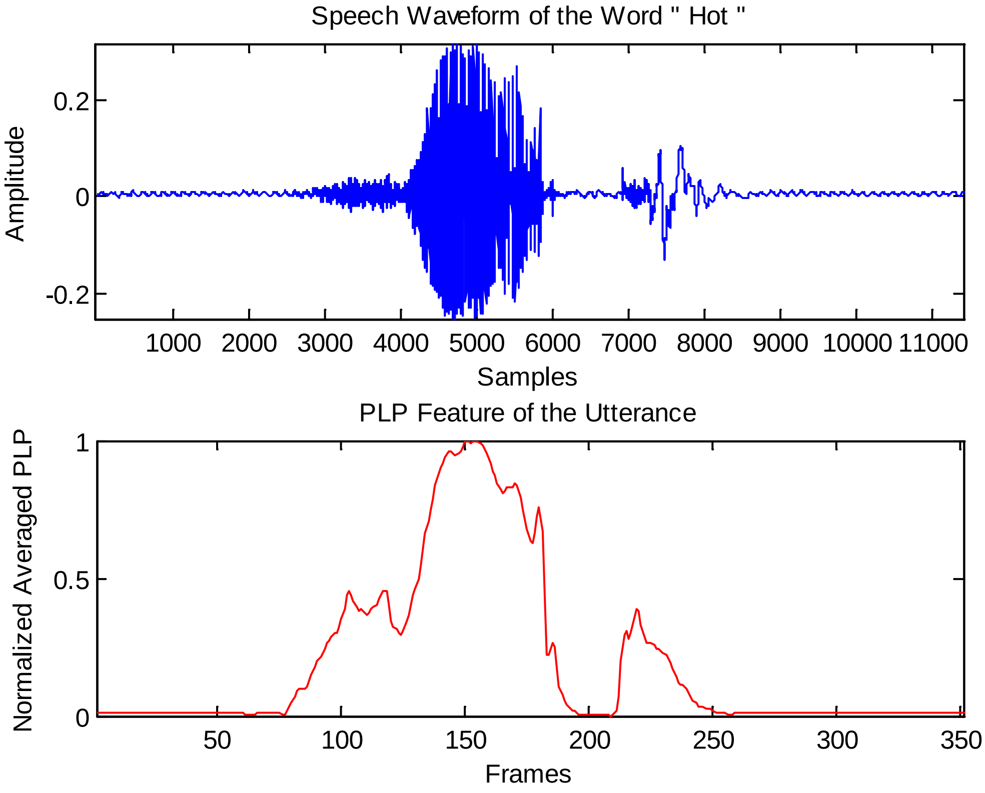


Figure 3.9: Waveform of the Word ”Hot”(Top Plot), and Corresponding PLP Curve (Bottom Plot).

3.4.3 Mel-Frequency Cepstral Coefficients

Cepstral analysis has been used extensively for feature extraction in speech recognition, and the most popular derivation of cepstral analysis combines the cepstrum with a nonlinear frequency-warping, known as the *Mel*-scale conversion. The resulting coefficients are called Mel-Frequency Cepstral Coefficients (MFCC). This technique has been first developed by Davis et al. in [27] .The basic idea of using MFCCs is to obtain a feature representation which approximates the behaviour of the auditory system. It adopts the characteristics of a human ear which is commonly assumed to be sensitive to the frequencies in the range (300Hz-3400Hz). Besides, the human auditory system is also known to have higher resolution in lower frequencies as compared to higher frequencies. For example, humans can easily discriminate between closely spaced low frequency tones such as 300 and 350 Hz, but not between closely spaced high frequency tones such as 3000 and 3050 Hz. As a result, the Mel scale maps an acoustic frequency to a perceptual frequency scale (as shown in Fig 3.10).

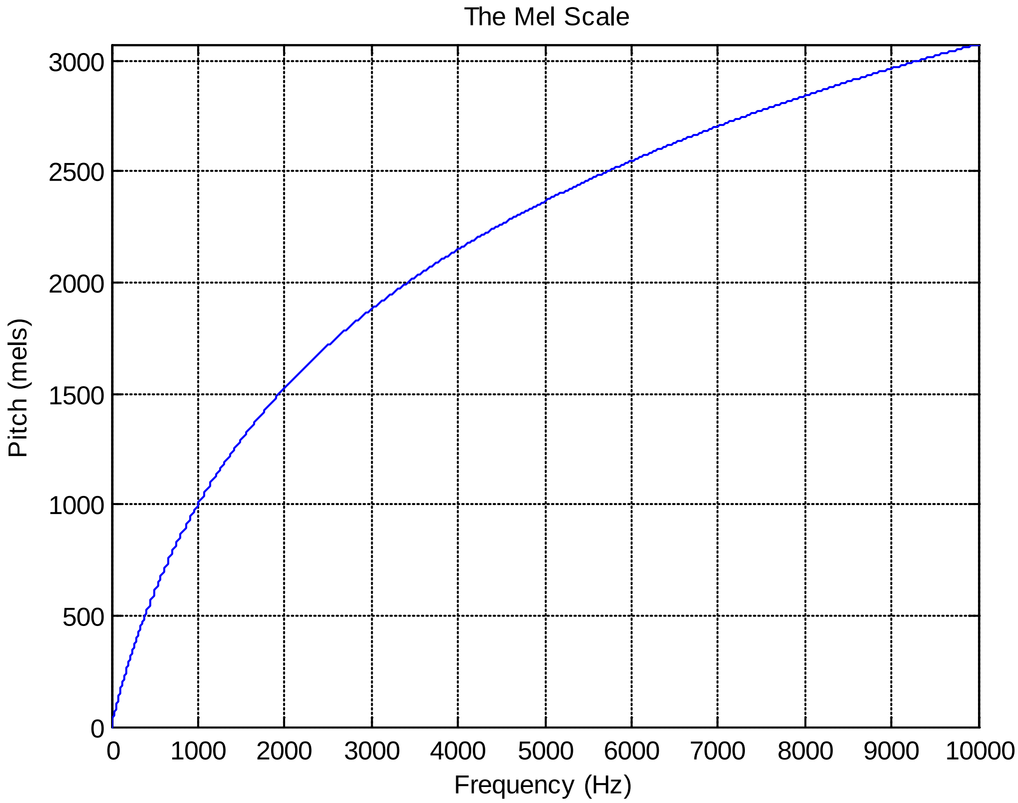


Figure 3.10: The Mel Scale.

Fig. 3.10 shows that the mapping is linear below 1 kHz while being logarithmic above 1 kHz, and thus it mimics the spectral characteristics of the human ear. Formally, the mapping is computed according to the following formula [24]:

$$F_{Mel} = 2595. log_{10}\left(1 + \frac{f(Hz)}{700}\right) \qquad (3.19)$$

One useful way to create Mel-spectrum is to use a filter bank, which uses one filter per a desired Mel-frequency component. Typically, each filter in this bank has a triangular band-pass frequency response. Such filters compute the average spectrum around each center frequency with increasing bandwidths, as displayed in Fig 3.11.

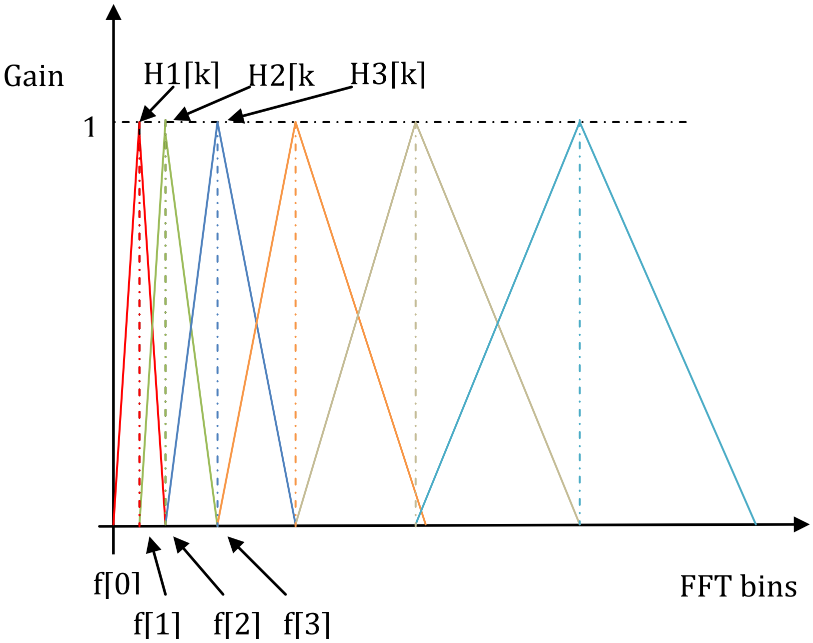


Figure 3.11: Triangular Filters Used to Compute MFCC.

The computation of MFCC can be summarised in the following steps:

1. The spectral energy of each pre-processed frame is computed as :

$$S_i = |S(k)|^2\,, \quad i = 0,1, \dots . N/2\,, \qquad (3.20)$$

where $S(k)$ is the N-point DFT of the frame defined as :

$$S(k) = \sum_{n=0}^{N-1} s(n)e^{-j2\pi kn/N}, \quad k = 0,1, \dots . N - 1 \qquad (3.21)$$

2. Next, the Mel-spectrum is obtained by multiplying the spectral energy of each band of the triangular Mel-weighting filters and integrating the result:

$$\tilde{S}_j = \sum_{i=0}^{N/2} S_i . H_j(i), \quad j = 0,1, \dots . , J - 1 \qquad (3.22)$$

where J is the total number of triangular Mel- weighting filters, $H_j(i)$, used.

3. Finally, the Mel-cepstrum (MFCC) is calculated by applying Discrete Cosine Transform (DCT) to the logarithm of the Mel-spectrum as follows:

$$c(n) = \sum_{j=0}^{J-1} log\left(\tilde{S}_j\right) \cos\left(\frac{\pi n}{2J}(2j+1)\right), \ n = 0,1, \dots . , C - 1 \qquad (3.23)$$

where C is the total number of cepstral coefficients.

Fig. 3.12 shows a block diagram of the computational process explained above.

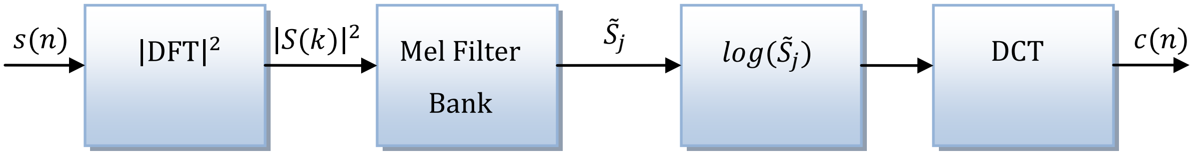


Figure 3.12: Block Diagram of MFCC Computation Process.

In this thesis, C has been set to be equal to 20. As a speech feature, the minimum value of $c(n)$ has been used. (As shown by the example of the word ”Hot” in Fig. 3.13).

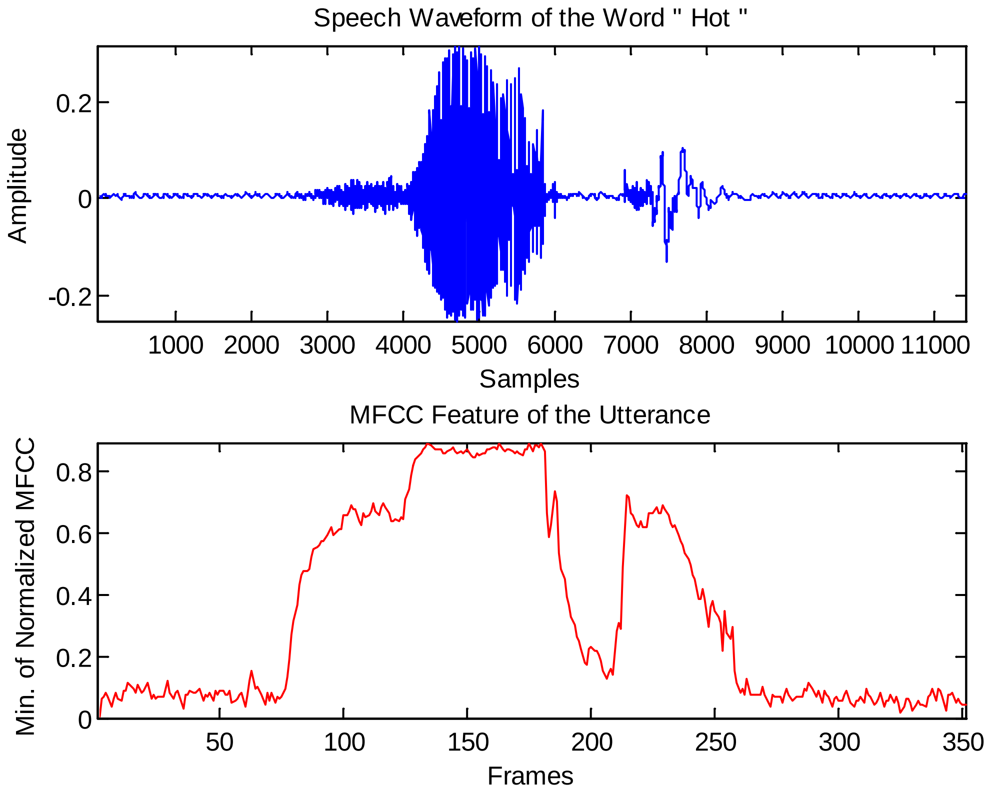


Figure 3.13: Waveform of the Word ”Hot”(Top Plot), and Corresponding MFCC Curve (Bottom Plot).

3.5 Speech Detection

Having all the speech features computed, the final module estimates the speech bounds. In this thesis we propose an original formulation of the speech segmentation procedure based on the approach first proposed in [28] in the field of image segmentation. This approach is commonly referred to as Chan-Vese model for active contours.

Chan-Vese model is a powerful and flexible method used in image processing to perform image segmentation, including some types of images that are difficult to segment based on their edges and/or using their histograms as done in [29, 30, 31, 32]. The model is based on an energy minimization problem, which can be efficiently solved using the level-set approach of [33].

This model is used in a wide range of applications, such as medical imaging [34], surveillance [35], robotics, control, just to name a few. In what follows, we provide a detailed description of our adaptation of this model and its assumptions to detect the speech endpoints successfully.

3.5.1 Problem Description

Let $\varphi(t)$, with $t \in \mathbb{R}$, be a 1-D analog of a level-set function in [33]. In this case:

$$\begin{array}{ll} \varphi(t) \leq 0 & , for\ t \in I \\ \varphi(t) > 0 & , for\ \ t \notin I \end{array} \quad , \qquad (3.24)$$

where I is a closed subset of $\mathbb{R}$, $I \subseteq \mathbb{R}$. In our case, we will use $I = [LB, RB]$, where LB and RB indicate the beginning and the ending of an utterance, respectively.

A particularly convenient way to obtain such a function is in the form of a *Signed Distance Function* (SDF). The Signed Distance Function can be obtained mathematically by negating the values in the range (LB, RB) of the Distance Function $\bar{\varphi}(t)$ defined by the following equation:

$$\bar{\varphi}(t) = min\{|t - LB|, |t - RB|\} \qquad (3.\ 25)$$

Fig. 3.14 shows the Distance Function and its corresponding SDF.

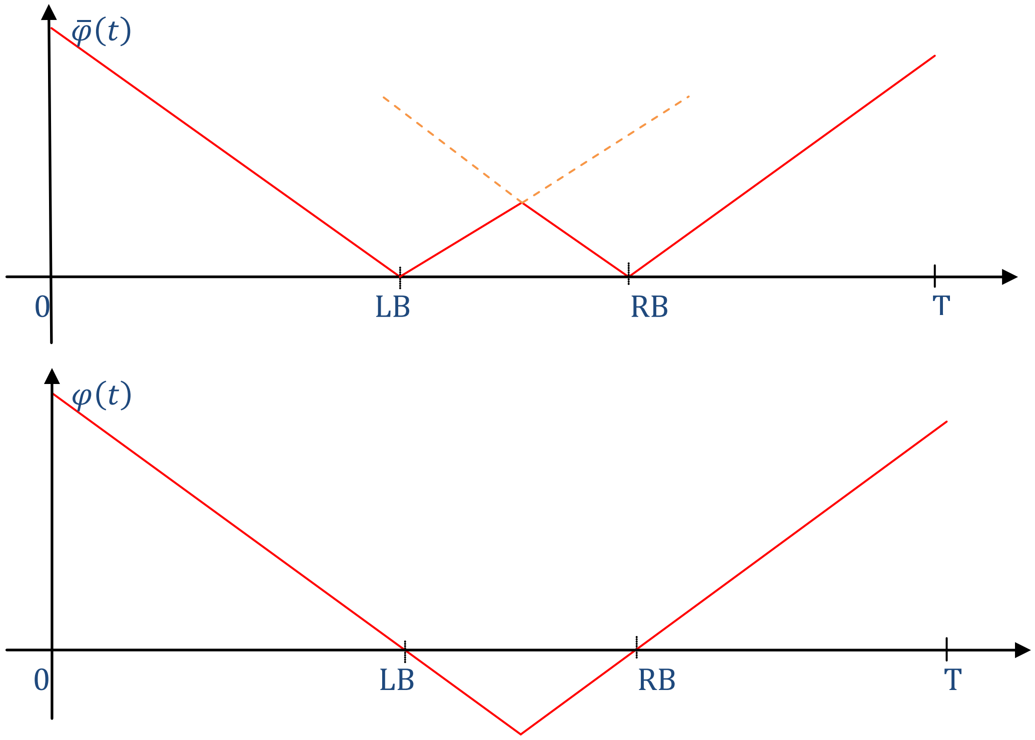


Figure 3.14: The Distance Function (Top Sketch), and its Corresponding Signed Distance Function (Bottom Sketch).

For practical reasons, instead of the whole $\mathbb{R}$, we work with a measurement interval $[0, T]$, and assume that $I \subset [0, T]$, and let I^C be the complement of I in $[0, T]$, i.e. $I^C = [0, T] \backslash I$.

Then, given a speech feature $u(t)$, we assume the latter to have different mean values over I and I^C. In this case, one could find an optimal I through solving the following optimization problem:

$$\xi\{I\} = \int_I (u(t) - \mu_{in})^2 \, dt + \int_{I^c} (u(t) - \mu_{out})^2 \, dt \qquad , \tag{3.26}$$

where

$$\mu_{in} = \int_{C_{in}} u(t)dt \Big/ \int_I \, dt \tag{3.27}$$

$$\mu_{out} = \int\limits_{C_{out}} u(t)dt \Bigg/ \int_{I^c} dt \qquad (3.28)$$

μ_{in} and μ_{out}denote the mean values of $u(t)$ over I and I^C , respectively.

In other words, we are looking for an optimal support I of the useful speech signal which satisfies:

$$I_{opt} = inf\{\xi(I)\} \qquad (3.\ 29)$$

where the minimization is performed over all closed subset of $[0, T]$.

Obviously, the above problem entails a combinatorial solution, which is undesirable from the practical point of view. A much more efficient formulation is possible in terms of the SDF $\varphi(t)$. In particular, let $H(t)$ be the Heaviside function defined by the following equation:

$$H(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases} \qquad (3.30)$$

Then, the minimization over I can be replaced by minimization over the SDF. In this case, the optimal $\varphi(t)$ should minimize the following cost function:

$$\xi\{\varphi(t)\} = \int\limits_0^T (u(t) - \mu_{in})^2\, H(-\varphi(t))dt + \int\limits_0^T (u(t) - \mu_{out})^2 H(\varphi(t))\, dt \qquad (3.31)$$

Note that $H(-\varphi(t))$ is essentially the indicator function of I.

In order to minimize $\xi\{\varphi(t)\}$ with respect to φ, we use the gradient-descent approach, which is given by a gradient flow equation of the form:

$$\frac{\partial \xi(\varphi,\tau)}{\partial \tau} = -\frac{\delta \xi(\varphi,\tau)}{\delta \varphi} \quad , \tag{3.32}$$

where τ is an artificial time variable(iteration), and $\delta\xi/\delta\varphi$ is the first variation of ξ(w.r.t. the level set function φ). The latter can be shown to be given by:

$$E\big(\varphi(t)\big) = \frac{\partial \xi\{\varphi(t)\}}{\partial \varphi} = \delta\big(\varphi(t)\big)[(u(t)-\mu_{out})^2 - (u(t)-\mu_{in})^2] \quad , \tag{3.33}$$

where $\delta(.)$ is the distributive derivative of $H(.)$, i.e. the Dirac delta function. In the discrete computation, we replace the partial derivative w.r.t. τ by its Euler approximation which results in:

$$\varphi^{n+1}(t) = \varphi^{n}(t) - \Delta_t E(\varphi^{n}(t)) \quad , \tag{3.34}$$

where Δ_t is a step size and n is an iteration index.

After every update of φ, we recomputed according to the following formulae:

$$\mu_{in} = \int_0^T u(t)H(-\varphi(t))dt \Bigg/ \int_0^T H(-\varphi(t))dt \tag{3. 35}$$

$$\mu_{out} = \int_0^T u(t)H(\varphi(t))dt \Bigg/ \int_0^T H(\varphi(t))dt \tag{3.36 }$$

The procedures of the proposed detection algorithm are further illustrated and summarized in subsection 3.5.4

3.5.2 Regularizations of the Heaviside and Dirac Delta Function

Using the Heaviside and delta functions in Equations (3.33), (3.35), and (3.36) is impossible for practical reasons. To alleviate this problem, we replace $H(t)$ and $\delta(t)$ by their smooth approximations (regularizations). Specifically, in this thesis we use:

$$H_\varepsilon(t) = \begin{cases} 1, & if\ t > \varepsilon \\ 0, & if\ t < -\varepsilon \\ \frac{1}{2}\left[1 + \frac{t}{\varepsilon} + \frac{1}{\pi} sin\left(\frac{\pi t}{\varepsilon}\right)\right], & if\ |t| \leq \varepsilon \end{cases} \tag{3.37}$$

$$\delta_\varepsilon(t) = H_\varepsilon{}'(t) = \begin{cases} 0, & if\ |t| > \varepsilon \\ \frac{1}{2\varepsilon}\left[1 + cos\left(\frac{\pi t}{\varepsilon}\right)\right], & if\ |t| < \varepsilon \end{cases} \tag{3.38 }$$

These approximations provide differentiability, resulting in a stable convergence, in the sense that they usually lead to a global minimum of the energy [28]. Fig. 3.15 shows the regularized Heaviside and Dirac delta functions with $\varepsilon = 1$.

3.5.3 Fusion of Different Features

In order to enhance the performance of the algorithm, multiple different features $\{u^k(t)\}_{k=1}^d$ can be fused, (In our thesis we set d to equal 3). In this case, the cost function and its first variation should be modified as given by:

$$\xi\{\varphi(t)\} = \sum_{k=1}^{d} \left\{ \int_0^T \left(u^k(t) - \mu^k{}_{in}\right)^2 H\left(-\varphi(t)\right)dt + \int_0^T (u^k(t) - \mu^k{}_{out})^2 H(\varphi(t))\, dt \right\} \tag{3.39}$$

$$E\left(\varphi(t)\right) = \frac{\partial \xi\{\varphi(t)\}}{\partial \varphi} = \ \delta\left(\varphi(t)\right). \sum_{k=1}^{d} \left\{(u^k(t) - \mu^k{}_{out})^2 - \left(u^k(t) - \mu^k{}_{in}\right)^2\right\} \tag{3.40}$$

Further, a set of weighting factors $\{\alpha_k\}_{k=1}^d$ can be assigned to indicate the importance of each feature compared to the others. Consequently, equation (3.40) can be modified to result in:

$$E\left(\varphi(t)\right) = \frac{\partial \xi\{\varphi(t)\}}{\partial \varphi} = \ \delta\left(\varphi(t)\right). \sum_{k=1}^{d} \alpha_k. \left\{(u^k(t) - \mu^k{}_{out})^2 - \left(u^k(t) - \mu^k{}_{in}\right)^2\right\} \tag{3. 41}$$

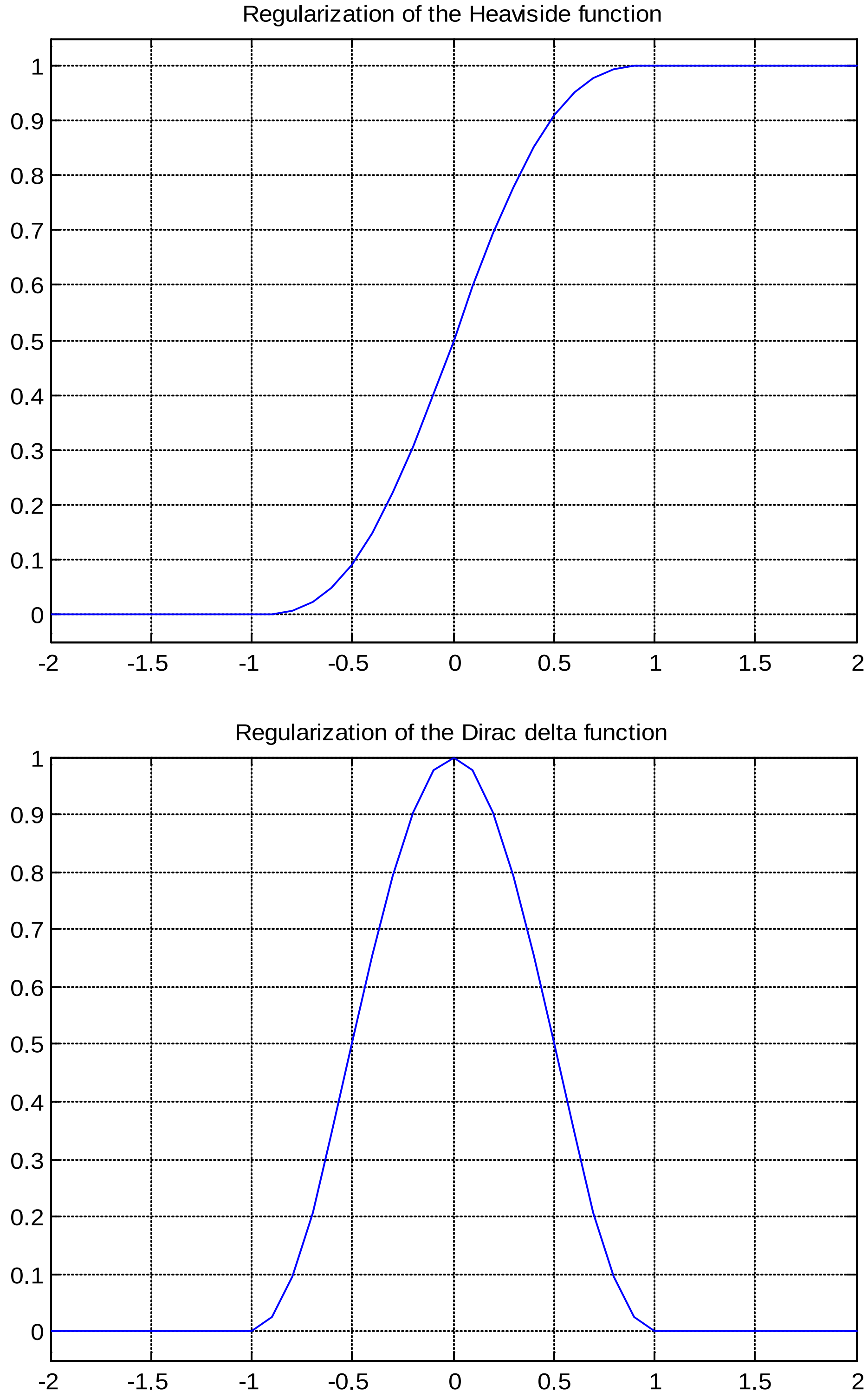


Figure 3.15: Regularized Heaviside and Dirac Delta Functions.

3.5.4 Summary of the Proposed Algorithm

1. Normalize the amplitude of the feature vectors to lie between 0 and 1, using an offset and linear scaling.
2. Initialize $\varphi^{n=0}$ to some signed distance function φ_0.
3. Compute μ_{in} and μ_{out} for the three features by using Equations (3.35) and (3.36), respectively.
4. Update φ^{n+1} by solving the PDE of Equation (3.34).
5. Reinitialize φ to be the signed distance function to $\{\varphi^{n+1} = 0\}$.(Optional step)
6. Check the stationarity of the solution. If it is not stationary, $n = n + 1$ and go to step (3), else stop. In practice, the process should be stopped when $\|\varphi^{n+1} - \varphi^{n}\| < \varepsilon$, where ε is an experimentally-set threshold, i.e., φ is not expected to change (except for some possible small numerical changes).

Once an optimal $\varphi(t)$ is computed, the associated support of the speech signal (utterance) can be recovered as:

$$I = \{t \mid \varphi(t) \leq 0\} \tag{3.42}$$

Chapter 4

Experimental Validation

4.1 Performance Analysis

To evaluate the performance of the proposed endpoint detection algorithm, two reference algorithms have been used for the purpose of comparative analysis. Specifically, the performance of the proposed algorithm is compared against that of the well-established and widely used algorithms of Rabiner and Sambur [5] and Huang and Yang [6]. Both the reference and the proposed algorithms have been applied to the same dataset under equivalent conditions (e.g., noise level, noise statistics, etc.). A quantitative comparison of the obtained results is presented in this chapter.

The speech segmentation boundaries, estimated using MATLAB by the proposed and reference algorithms, were compared against the boundaries obtained via manual segmentation by skilled personnel. As a quantitative comparison metric we have used the relative amount by which the detected boundary $\hat{x}$ differs from its corresponding manual value x. Formally, the Mean Squared Error (MSE) is defined as given by:

$$MSE = \frac{1}{N} . \sum_{i=1}^{N} e(i)^2 = \frac{1}{N} . \sum_{i=1}^{N} (x(i) - \hat{x}(i))^2 \tag{4.1}$$

where *N* is the number of estimates.

4.1.1 Support Database

The experimental results of this thesis have been obtained based on the dataset provided through *Support Database* [36], which is known to represent nearly all phonetic sounds of American English. This database contains hard types of phonemes to endpoint detection algorithms such as *stops*, *frictions (fricatives)*, *glides* and *nasals* due to their low energy content. So, using this database, we have been able to test the performance of our method for a broad range of the different types of phonemes.

The database exploited in this study was composed of 68 audio recordings of the voice of a female speaker. Each recording consisted of between 3 to 6 words in addition to the sound of the phoneme represented by the given recording. All the recordings under processing have been acquired in a studio (i.e., clear signal).

All the data sequences have been modified in the way described in [37]. Specifically, all the data file were first converted to the WAV (Waveform Audio File) format. Subsequently, the data sequences were down-sampled to the Nyquist rate of 4 kHz, followed by quantization to 16 bits resolution. Finally, all the recordings were segmented into individual words (with the exclusion of pure phonemes), resulting in a total of 326 audio signals.

4.1.2 Experimental Results

A series of experiments have been carried out to analyze the performance of the proposed and the reference algorithms under different types of measurement conditions. Specifically, as an initial step, all the algorithms were applied to noise-free data. Table 4.1 summarized the resulting errors obtained by different methods under comparison. One can clearly see that, in terms of the mean squared error (MSE), the proposed algorithm outperforms the reference methods in detecting both the starting and the ending points of the speech segments. The "Overall MSE" column of Table 4.1, which is the average of the starting and the ending point MSE, presents the error reduction by 1.1% and 4.8% when compared to Rabiner and Huang algorithms, respectively.

Table 4.1: MSE for various algorithms under comparison (SNR = Inf dB).

Method	Starting Point MSE	Ending Point MSE	Overall MSE
Algorithm	1.9%	0.7%	1.3%
Huang	6.3%	5.8%	6.1%
Rabiner	2.1%	2.8%	2.4%

As the next step, the data signals were contaminated by various levels of three different types of measurement noises, viz. by white, pink, and car (Volvo) noise. All the noises have been taken from the NOISEX-92 database [38], followed by scaling their amplitude to set up a required value of SNR in the range from 0dB to 30dB.

The results obtained in the case of white noise contamination are summarized in Fig. 4.1. Our algorithm is observably superior to the other algorithms, especially at low SNRs when detecting the starting points and at high SNRs when detecting the ending points. The overall error has also been measured and visualized in Fig. 4.1. It can be seen that our algorithm outperforms the competitive algorithms at all SNR levels.

Figures 4.2 and 4.3 show the results of the conducted experiments in case of pink noise and car noise, respectively. As evident from figures, our algorithm outperforms the reference methods, especially in detecting the ending points. As one can see, the proposed algorithm has the best overall MSE values, indicating a better performance of all compared endpoint detection methods.

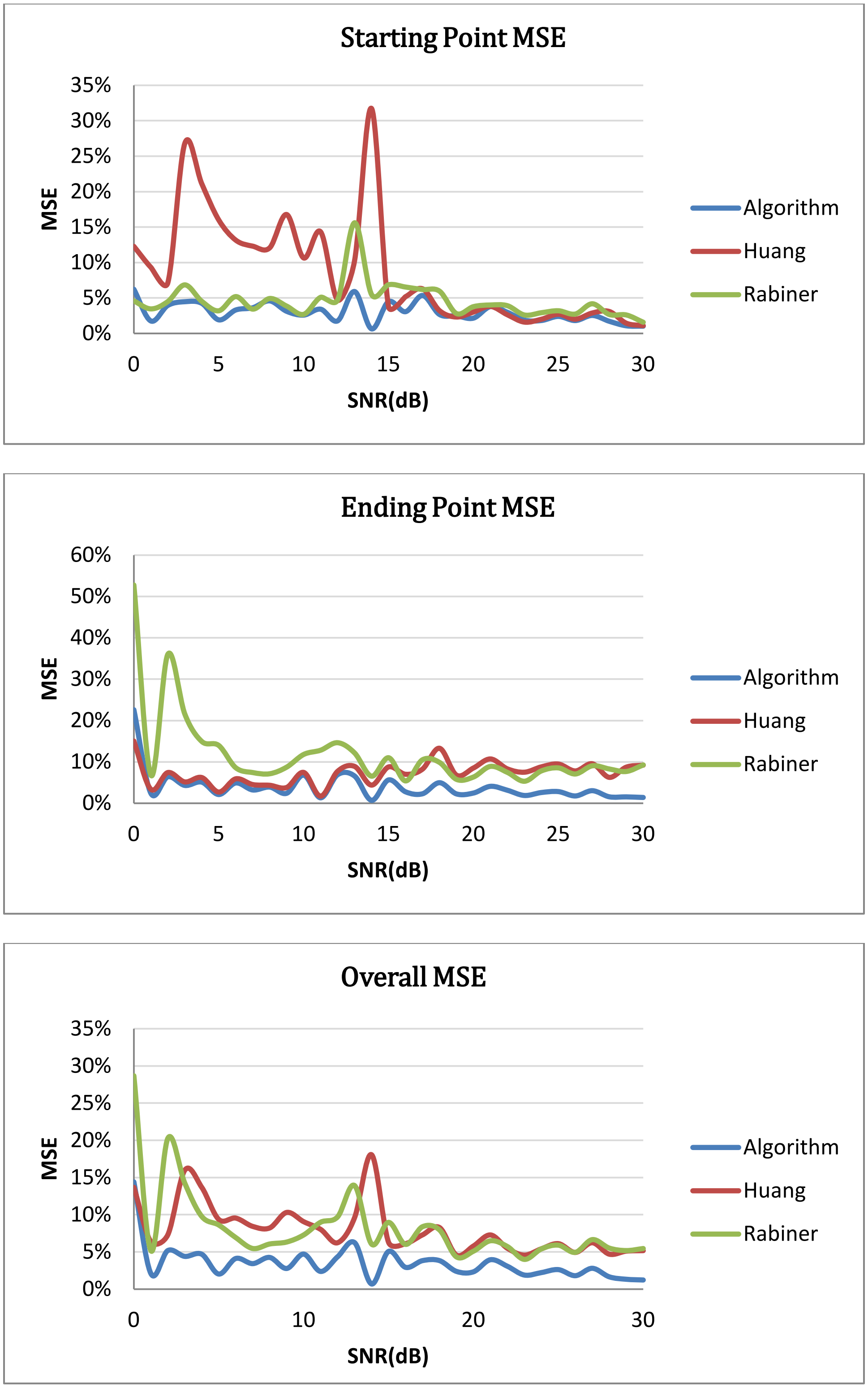
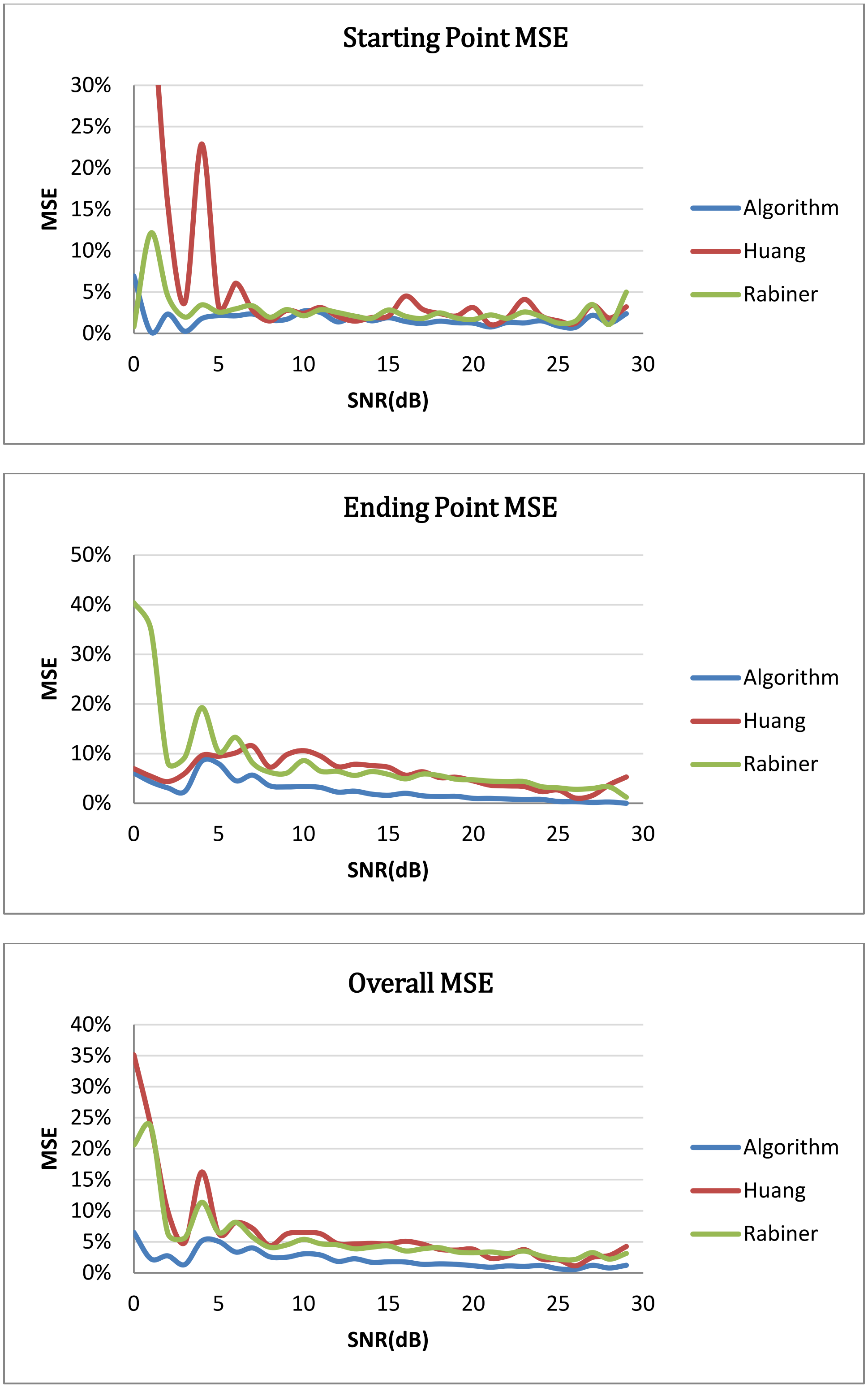


Figure 4.1: MSE of the proposed algorithm against the two reference algorithms with white noise added.



Starting Point MSE

Ending Point MSE

Overall MSE

Figure 4.2: MSE of the proposed algorithm against the two reference algorithms with pink noise added.

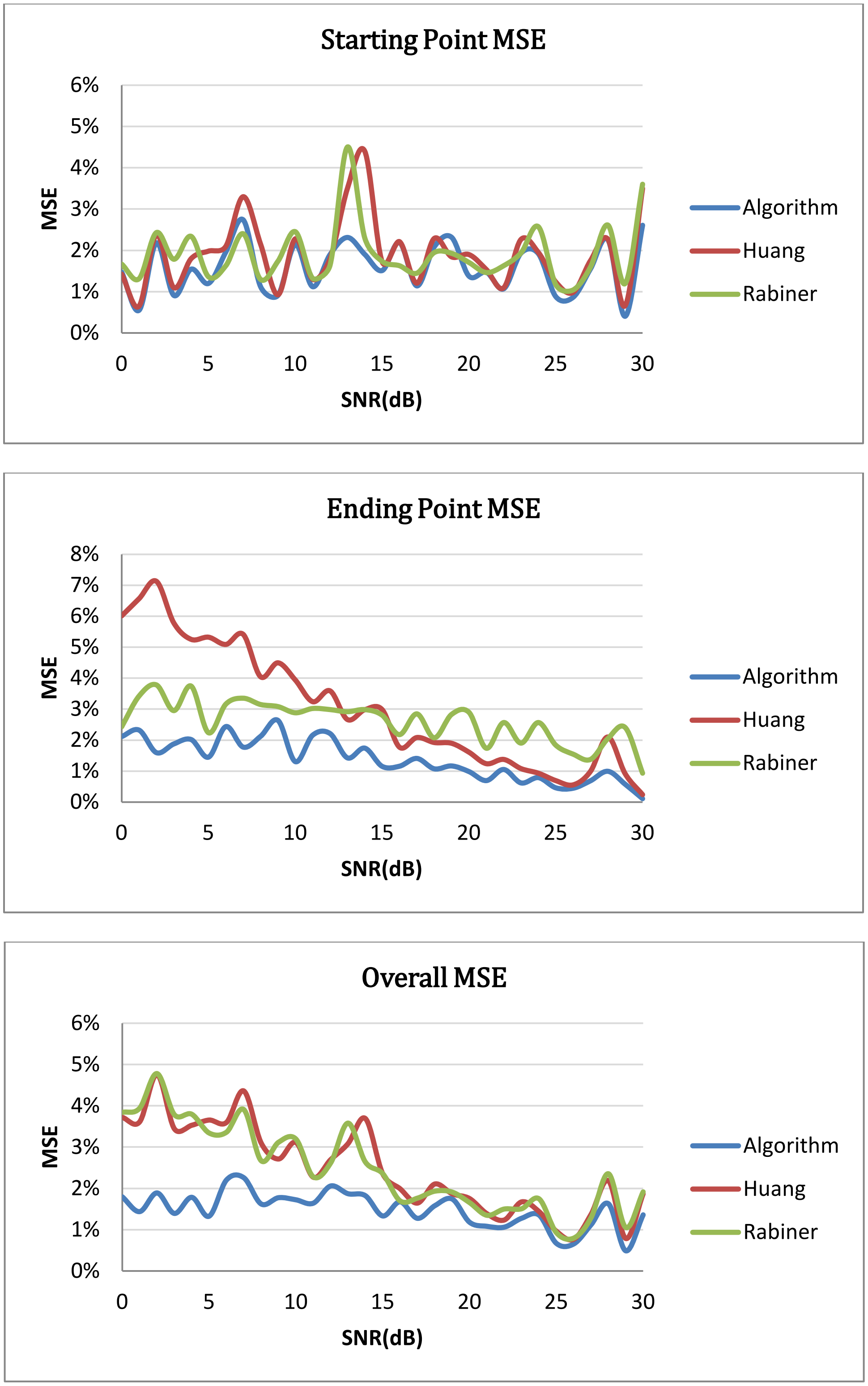


Figure 4.3: MSE of the proposed algorithm against the two reference algorithms with car noise added.

4.2 Multi-phoneme Comparison between Algorithms

This section presents the results of a comparison between the three algorithms in terms of their mean square error (MSE) as they are applied to 14 basic types of American English phonemes: *3-element blend, affricate, back vowel, central vowel, diphthong, friction, front vowel, glide, l-blend, liquid, nasal, r-blend, s-blend,* and *stop* sounds. The tests were performed in case of clean speech and with addition of the artificial noise (white, pink and car) as in the previous section.

Results of the tests in case of noise-free speech can be found in Table 4.2 and Fig. 4.4 which translates the numerical results into graphical ones. We can see that the proposed algorithm provides more accurate estimation results than the reference algorithms, except in the case of detecting the starting point in *stops*, and the ending points in *back vowels*.

Conducting tests in the presence of white noise resulted in the MSE values shown in Tables 4.3, 4.4, and 4.5, and their corresponding Figures 4.5, 4.6, and 4.7, which describe the behaviour of the proposed algorithm and other algorithms under three groups of SNRs. The first group represents low SNR (0dB≤ SNR<10dB), for which the performance of the methods are illustrated in Table 4.3 and Fig. 4.5. The results clearly show that the proposed algorithm results in the lowest overall MSE, as well as show an acceptable performance when detecting the starting and ending points, except at the *s-blends* and the*3–element blend*s, respectively.

Least fortunate performance of the proposed algorithm is observed on the second group (10dB≤ SNR<20dB) are at the *back vowels, frictions, l-blends* and the *nasals* when detecting the starting points and the *central vowels* when detecting the ending points. However, Fig. 4.6 and Table 4.4 demonstrate that the superior performance of the proposed algorithm on the rest of the phonemes as well as the enhancement of the overall error.

As can be seen from Fig. 4.7 and Table 4.5, which represent the case of SNR equal to (20dB≤ SNR≤30dB), the results are the best among the three groups of SNR, since there was only one phoneme "*diphthongs"* at which Huang's algorithm was able to produce lower MSE than ours when detecting the starting points.

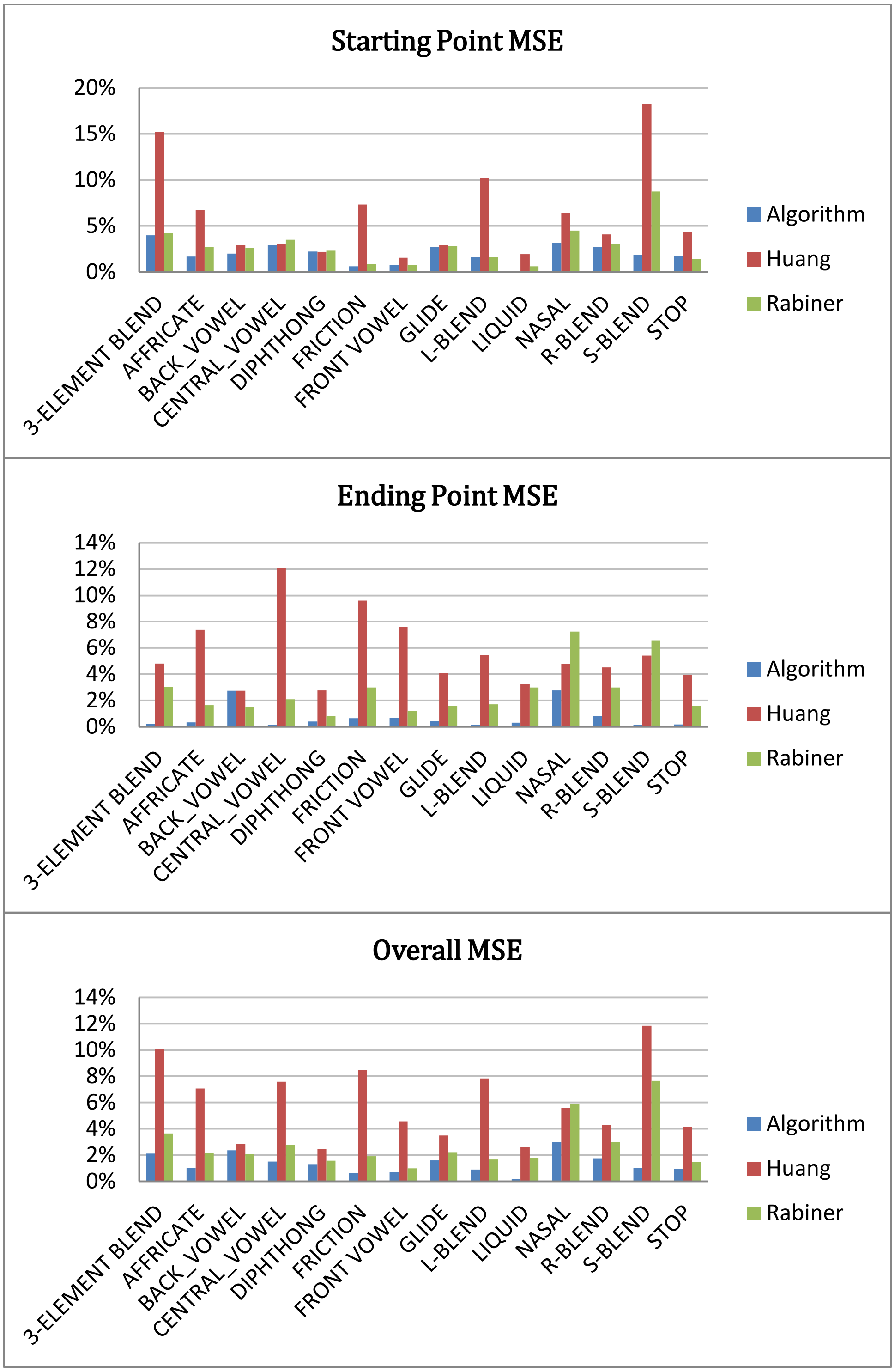
For the case of pink noise, the results are summarized in Tables 4.6, 4.7 and 4.8 and their visualized Figures 4.8, 4.9, and 4.10 for the three groups of SNRs mentioned earlier. From theses tables and figures one can see that the proposed algorithm still outperforms the reference algorithms in almost each group of SNR and in almost each phoneme, with the following exceptions:

1. At low SNRs, the proposed approach has a slightly higher MSE than Rabiner's algorithm when detecting the ending points of the *diphthongs*. While when detecting the starting points at medium SNRs, the MSE of *diphthongs* and *stops* were not the least among the others.
2. Also, the *glide* sounds of starting points of the last group of SNRs (high SNRs) are the only phoneme at which the proposed algorithms relinquish the lead.

Performing the same experiments in case of car noise resulted in the comparative metrics shown in Figures 4.11, 4.12 and 4.13 and their detailed Tables 4.9, 4.10 and 4.11. If we inspect them, we can find that our algorithm maintains the high performance as obtained with the other two types of noises. However, some exceptions can be found in each group of SNRs. At low SNRs, our algorithm are not the best when detecting *affricates, frictions* and *r-blends* of the starting points as well as the *diphthongs* of the ending points. Also at medium SNRs, *central vowels, glides* and *r-blends* of the starting points and *affricates* of the ending points are the cases at which MSE of the proposed algorithm performs worse as compared to the other algorithms. The last group of SNR experiments shows that *friction* sounds of the starting points and *affricates, r-blends* and *stops* of the ending points are the weaknesses of our algorithm.

Table 4.2: MSE per phoneme type, in case of clean speech.

Phoneme Type	Method	Mean Square Error		
		Starting Point	Ending Point	Overall
3-ELEMENT BLEND	Algorithm	4.0%	0.2%	2.1%
	Huang	15.2%	4.8%	10.0%
	Rabiner	4.3%	3.0%	3.6%
AFFRICATE	Algorithm	1.7%	0.3%	1.0%
	Huang	6.7%	7.4%	7.1%
	Rabiner	2.7%	1.6%	2.2%
BACK VOWEL	Algorithm	2.0%	**2.7%**	2.4%
	Huang	2.9%	2.7%	2.8%
	Rabiner	2.6%	1.5%	2.1%
CENTRAL VOWEL	Algorithm	2.9%	0.1%	1.5%
	Huang	3.1%	12.1%	7.6%
	Rabiner	3.5%	2.1%	2.8%
DIPHTHONG	Algorithm	2.2%	0.4%	1.3%
	Huang	2.2%	2.8%	2.5%
	Rabiner	2.3%	0.8%	1.6%
FRICTION	Algorithm	0.6%	0.7%	0.6%
	Huang	7.3%	9.6%	8.5%
	Rabiner	0.8%	3.0%	1.9%
FRONT VOWEL	Algorithm	0.8%	0.7%	0.7%
	Huang	1.5%	7.6%	4.6%
	Rabiner	0.8%	1.2%	1.0%
GLIDE	Algorithm	2.7%	0.4%	1.6%
	Huang	2.9%	4.1%	3.5%
	Rabiner	2.8%	1.6%	2.2%
L-BLEND	Algorithm	1.6%	0.2%	0.9%
	Huang	10.2%	5.4%	7.8%
	Rabiner	1.6%	1.7%	1.7%
LIQUID	Algorithm	0.0%	0.3%	0.2%
	Huang	1.9%	3.2%	2.6%
	Rabiner	0.6%	3.0%	1.8%
NASAL	Algorithm	3.2%	2.8%	3.0%
	Huang	6.4%	4.8%	5.6%
	Rabiner	4.5%	7.2%	5.9%
R-BLEND	Algorithm	2.7%	0.8%	1.8%
	Huang	4.1%	4.5%	4.3%
	Rabiner	3.0%	3.0%	3.0%
S-BLEND	Algorithm	1.9%	0.2%	1.0%
	Huang	18.3%	5.4%	11.8%
	Rabiner	8.7%	6.6%	7.6%
STOP	Algorithm	**1.7%**	0.2%	1.0%
	Huang	4.3%	4.0%	4.1%
	Rabiner	1.4%	1.6%	1.5%


Starting Point MSE

Ending Point MSE

Overall MSE

Figure 4.4: MSE per phoneme type, in case of clean speech.

Table 4.3: MSE per phoneme type, in case of white noise (0dB ≤SNR<10dB).

Phoneme Type	Method	Mean Square Error		
		Starting Point	Ending Point	Overall
3-ELEMENT BLEND	Algorithm	9.5%	**6.6%**	8.0%
	Huang	17.0%	7.8%	12.4%
	Rabiner	9.6%	6.5%	8.0%
AFFRICATE	Algorithm	1.0%	4.9%	3.0%
	Huang	1.2%	4.9%	3.0%
	Rabiner	4.2%	5.1%	4.7%
BACK VOWEL	Algorithm	2.0%	11.3%	6.7%
	Huang	20.7%	13.0%	16.8%
	Rabiner	5.7%	29.6%	17.6%
CENTRAL VOWEL	Algorithm	1.0%	6.2%	3.6%
	Huang	9.2%	6.2%	7.7%
	Rabiner	1.9%	6.8%	4.4%
DIPHTHONG	Algorithm	1.2%	4.1%	2.6%
	Huang	10.3%	4.2%	7.2%
	Rabiner	5.5%	13.4%	9.5%
FRICTION	Algorithm	4.1%	11.5%	7.8%
	Huang	17.5%	13.2%	15.3%
	Rabiner	4.2%	13.9%	9.0%
FRONT VOWEL	Algorithm	2.3%	4.7%	3.5%
	Huang	9.4%	5.0%	7.2%
	Rabiner	3.4%	8.1%	5.7%
GLIDE	Algorithm	3.6%	8.6%	6.1%
	Huang	13.5%	9.3%	11.4%
	Rabiner	3.8%	18.3%	11.0%
L-BLEND	Algorithm	1.2%	11.4%	6.3%
	Huang	5.8%	12.4%	9.1%
	Rabiner	1.5%	13.5%	7.5%
LIQUID	Algorithm	4.5%	11.6%	8.1%
	Huang	16.6%	12.4%	14.5%
	Rabiner	4.5%	13.4%	9.0%
NASAL	Algorithm	2.1%	9.8%	5.9%
	Huang	9.7%	10.0%	9.9%
	Rabiner	2.1%	18.8%	10.4%
R-BLEND	Algorithm	3.7%	9.6%	6.6%
	Huang	15.0%	9.6%	12.3%
	Rabiner	3.8%	9.8%	6.8%
S-BLEND	Algorithm	**12.9%**	8.2%	10.5%
	Huang	23.4%	8.7%	16.1%
	Rabiner	10.3%	17.5%	13.9%
STOP	Algorithm	1.3%	6.3%	3.8%
	Huang	23.1%	8.4%	15.7%
	Rabiner	1.8%	7.0%	4.4%

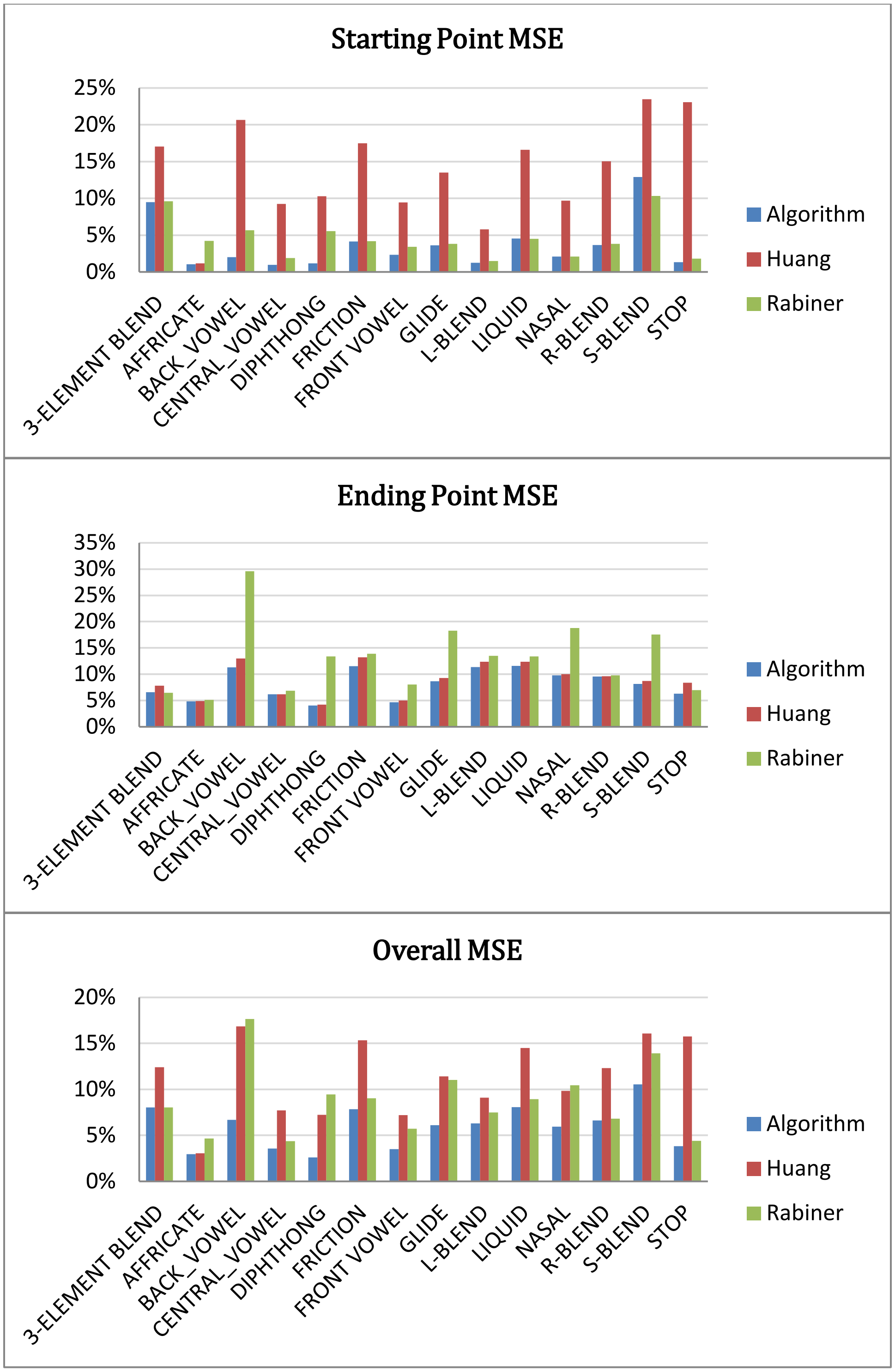


Figure 4.5: MSE per phoneme type, in case of white noise (0dB ≤SNR<10dB).

Table 4.4: MSE per phoneme type, in case of white noise (10dB ≤SNR<20dB).

Phoneme Type	Method	Mean Square Error		
		Starting Point	Ending Point	Overall
3-ELEMENT BLEND	Algorithm	3.0%	1.8%	2.4%
	Huang	10.6%	3.3%	6.9%
	Rabiner	9.9%	3.6%	6.7%
AFFRICATE	Algorithm	3.1%	2.4%	2.8%
	Huang	6.0%	6.2%	6.1%
	Rabiner	3.2%	6.5%	4.9%
BACK VOWEL	Algorithm	**4.2%**	3.0%	3.6%
	Huang	2.9%	8.3%	5.6%
	Rabiner	5.1%	3.8%	4.4%
CENTRAL VOWEL	Algorithm	0.7%	**5.1%**	2.9%
	Huang	2.2%	9.7%	5.9%
	Rabiner	1.0%	5.0%	3.0%
DIPHTHONG	Algorithm	1.0%	0.8%	0.9%
	Huang	1.4%	6.3%	3.8%
	Rabiner	1.2%	0.8%	1.0%
FRICTION	Algorithm	**6.4%**	12.3%	9.4%
	Huang	17.6%	15.2%	16.4%
	Rabiner	6.3%	15.3%	10.8%
FRONT VOWEL	Algorithm	0.5%	2.8%	1.6%
	Huang	2.2%	3.3%	2.7%
	Rabiner	0.6%	7.0%	3.8%
GLIDE	Algorithm	1.9%	2.8%	2.3%
	Huang	2.9%	3.9%	3.4%
	Rabiner	2.1%	3.4%	2.7%
L-BLEND	Algorithm	**1.1%**	5.5%	3.3%
	Huang	2.8%	9.3%	6.0%
	Rabiner	1.0%	7.6%	4.3%
LIQUID	Algorithm	0.4%	3.2%	1.8%
	Huang	0.5%	13.1%	6.8%
	Rabiner	3.3%	9.6%	6.4%
NASAL	Algorithm	**5.8%**	7.1%	6.5%
	Huang	5.3%	9.9%	7.6%
	Rabiner	10.7%	12.9%	11.8%
R-BLEND	Algorithm	2.4%	2.4%	2.4%
	Huang	2.6%	8.8%	5.7%
	Rabiner	2.5%	9.7%	6.1%
S-BLEND	Algorithm	2.3%	4.3%	3.3%
	Huang	3.4%	5.6%	4.5%
	Rabiner	11.7%	15.3%	13.5%
STOP	Algorithm	1.2%	2.6%	1.9%
	Huang	5.6%	3.3%	4.4%
	Rabiner	1.3%	7.0%	4.2%

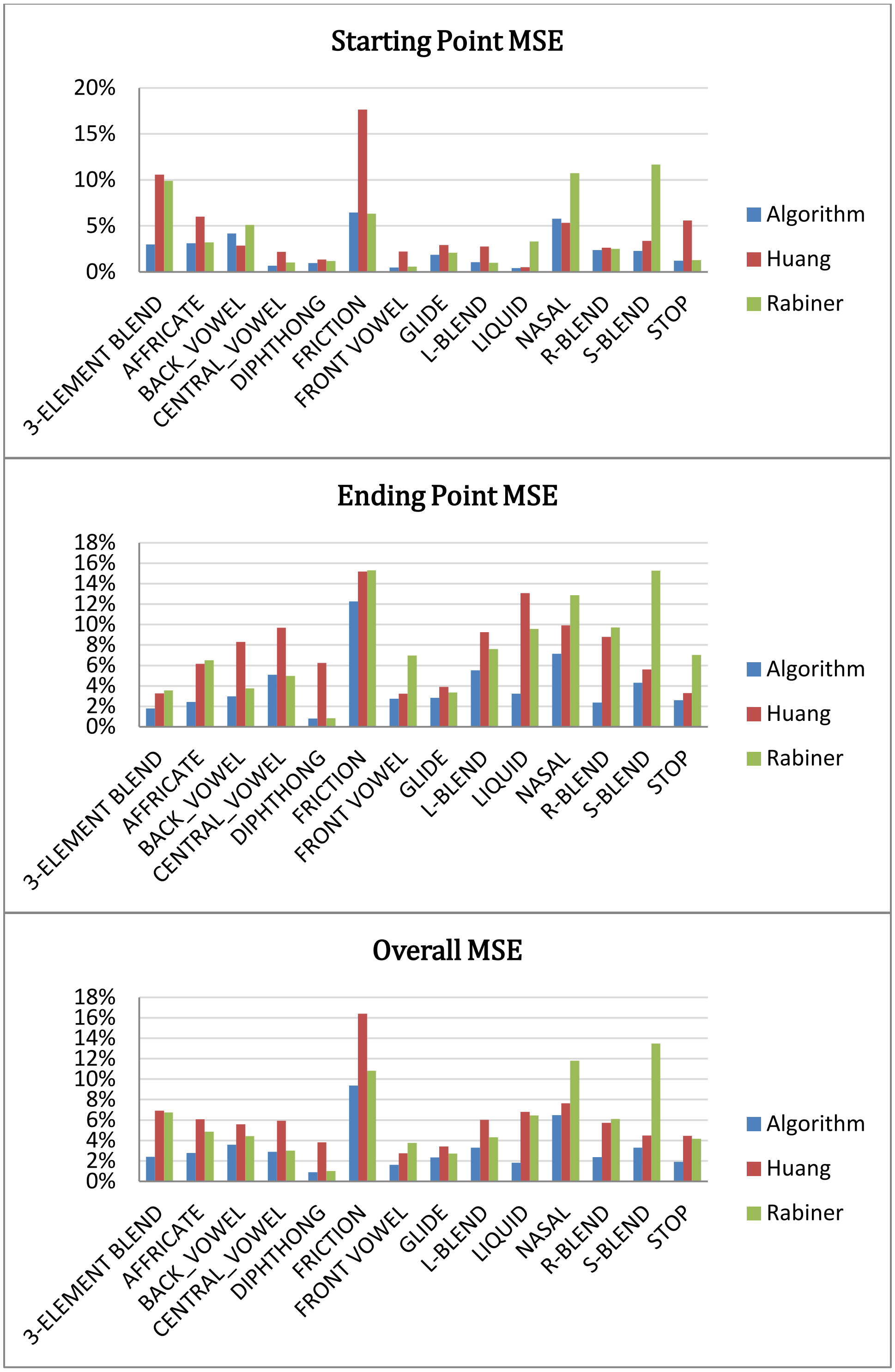


Figure 4.6: MSE per phoneme type, in case of white noise (10dB ≤SNR<20dB).

Table 4.5: MSE per phoneme type, in case of white noise (20dB ≤SNR<30dB).

Phoneme Type	Method	Mean Square Error		
		Starting Point	Ending Point	Overall
3-ELEMENT BLEND	Algorithm	1.5%	1.0%	1.2%
	Huang	2.5%	8.5%	5.5%
	Rabiner	8.3%	5.3%	6.8%
AFFRICATE	Algorithm	1.2%	1.0%	1.1%
	Huang	1.3%	4.1%	2.7%
	Rabiner	1.8%	3.1%	2.4%
BACK VOWEL	Algorithm	3.2%	2.9%	3.0%
	Huang	3.2%	7.9%	5.5%
	Rabiner	3.5%	7.4%	5.4%
CENTRAL VOWEL	Algorithm	3.2%	1.6%	2.4%
	Huang	3.5%	6.4%	4.9%
	Rabiner	3.6%	6.1%	4.8%
DIPHTHONG	Algorithm	**1.7%**	1.2%	1.5%
	Huang	1.6%	4.0%	2.8%
	Rabiner	1.8%	2.8%	2.3%
FRICTION	Algorithm	4.4%	6.6%	5.5%
	Huang	5.0%	16.1%	10.5%
	Rabiner	5.4%	15.2%	10.3%
FRONT VOWEL	Algorithm	1.3%	2.2%	1.7%
	Huang	1.3%	6.2%	3.8%
	Rabiner	1.7%	9.3%	5.5%
GLIDE	Algorithm	1.0%	1.1%	1.0%
	Huang	6.7%	8.7%	7.7%
	Rabiner	1.4%	6.4%	3.9%
L-BLEND	Algorithm	1.1%	0.5%	0.8%
	Huang	1.1%	7.6%	4.3%
	Rabiner	1.2%	4.3%	2.8%
LIQUID	Algorithm	0.3%	0.9%	0.6%
	Huang	0.4%	7.4%	3.9%
	Rabiner	1.7%	5.2%	3.4%
NASAL	Algorithm	3.0%	7.6%	5.3%
	Huang	3.6%	14.4%	9.0%
	Rabiner	3.3%	15.4%	9.3%
R-BLEND	Algorithm	2.3%	2.3%	2.3%
	Huang	3.0%	8.4%	5.7%
	Rabiner	2.7%	8.2%	5.5%
S-BLEND	Algorithm	2.4%	1.4%	1.9%
	Huang	3.6%	6.1%	4.9%
	Rabiner	11.8%	5.3%	8.5%
STOP	Algorithm	1.4%	0.4%	0.9%
	Huang	1.7%	6.7%	4.2%
	Rabiner	1.8%	6.4%	4.1%

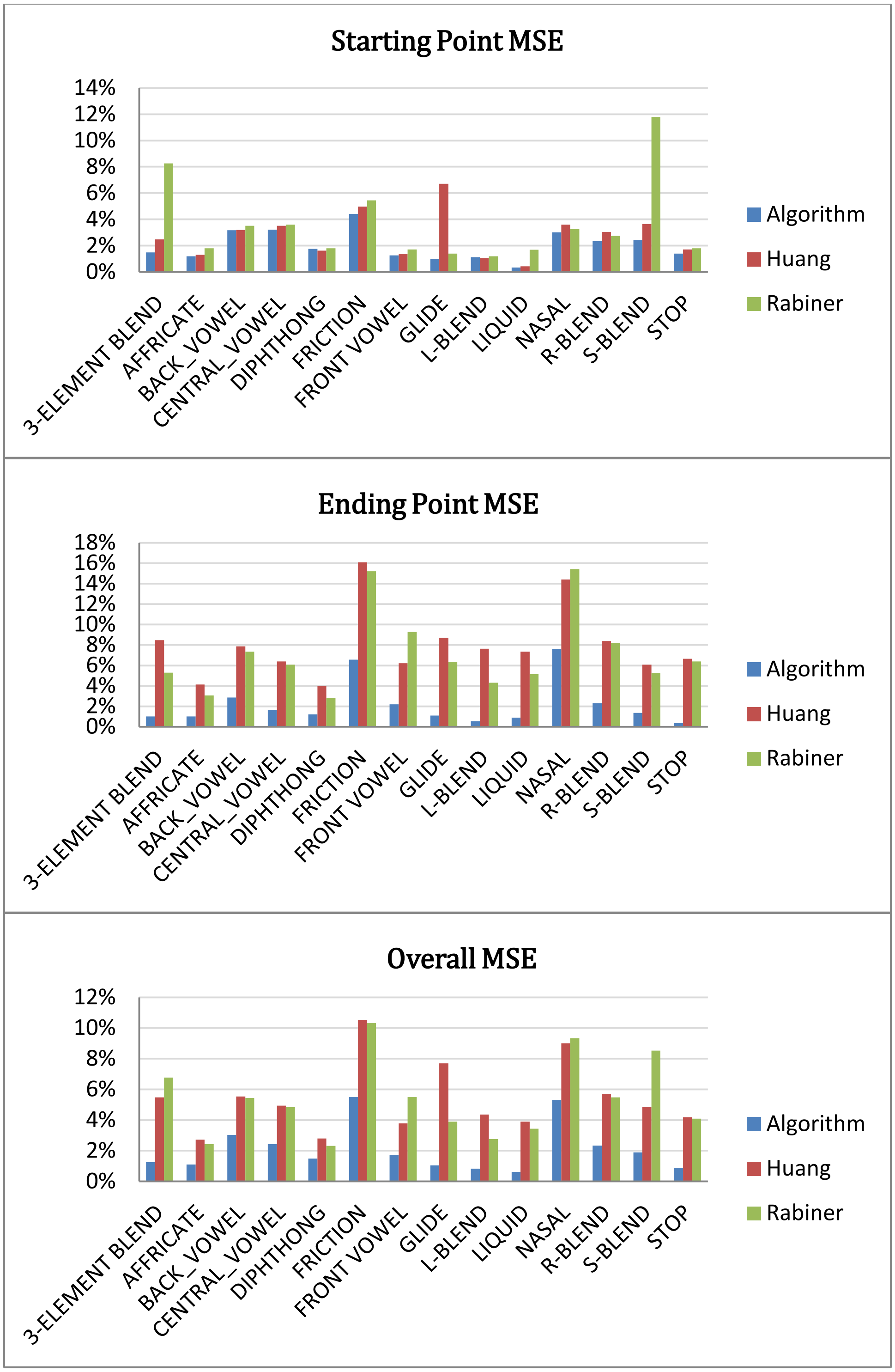


Figure 4.7: MSE per phoneme type, in case of white noise (20dB≤ SNR≤30dB).

Table 4.6: MSE per phoneme type, in case of pink noise (0dB≤ SNR<10dB).

	Method	Mean Square Error		
Phoneme Type		Starting Point	Ending Point	Overall
3-ELEMENT BLEND	Algorithm	5.8%	3.4%	4.6%
	Huang	10.4%	7.7%	9.0%
	Rabiner	9.0%	3.4%	6.2%
AFFRICATE	Algorithm	0.4%	4.3%	2.4%
	Huang	0.5%	8.6%	4.5%
	Rabiner	0.9%	5.9%	3.4%
BACK VOWEL	Algorithm	1.8%	4.1%	2.9%
	Huang	1.9%	9.1%	5.5%
	Rabiner	2.1%	6.1%	4.1%
CENTRAL VOWEL	Algorithm	1.1%	4.8%	2.9%
	Huang	1.8%	9.3%	5.5%
	Rabiner	1.5%	4.8%	3.1%
DIPHTHONG	Algorithm	1.0%	**2.2%**	1.6%
	Huang	1.2%	6.7%	3.9%
	Rabiner	1.3%	1.5%	1.4%
FRICTION	Algorithm	2.1%	5.0%	3.6%
	Huang	14.6%	8.9%	11.7%
	Rabiner	2.4%	14.6%	8.5%
FRONT VOWEL	Algorithm	2.3%	4.7%	3.5%
	Huang	2.4%	9.6%	6.0%
	Rabiner	2.9%	7.0%	4.9%
GLIDE	Algorithm	1.5%	3.6%	2.6%
	Huang	1.7%	10.6%	6.2%
	Rabiner	2.3%	6.5%	4.4%
L-BLEND	Algorithm	0.7%	4.5%	2.6%
	Huang	1.2%	18.6%	9.9%
	Rabiner	0.8%	6.6%	3.7%
LIQUID	Algorithm	1.6%	2.5%	2.1%
	Huang	1.9%	12.6%	7.2%
	Rabiner	3.7%	4.3%	4.0%
NASAL	Algorithm	2.4%	9.9%	6.2%
	Huang	6.5%	10.4%	8.5%
	Rabiner	2.4%	12.1%	7.2%
R-BLEND	Algorithm	1.0%	4.2%	2.6%
	Huang	1.7%	8.4%	5.0%
	Rabiner	1.0%	8.0%	4.5%
S-BLEND	Algorithm	2.8%	7.7%	5.2%
	Huang	11.3%	9.2%	10.3%
	Rabiner	10.9%	17.6%	14.2%
STOP	Algorithm	1.0%	1.1%	1.0%
	Huang	1.6%	5.6%	3.6%
	Rabiner	1.0%	5.6%	3.3%

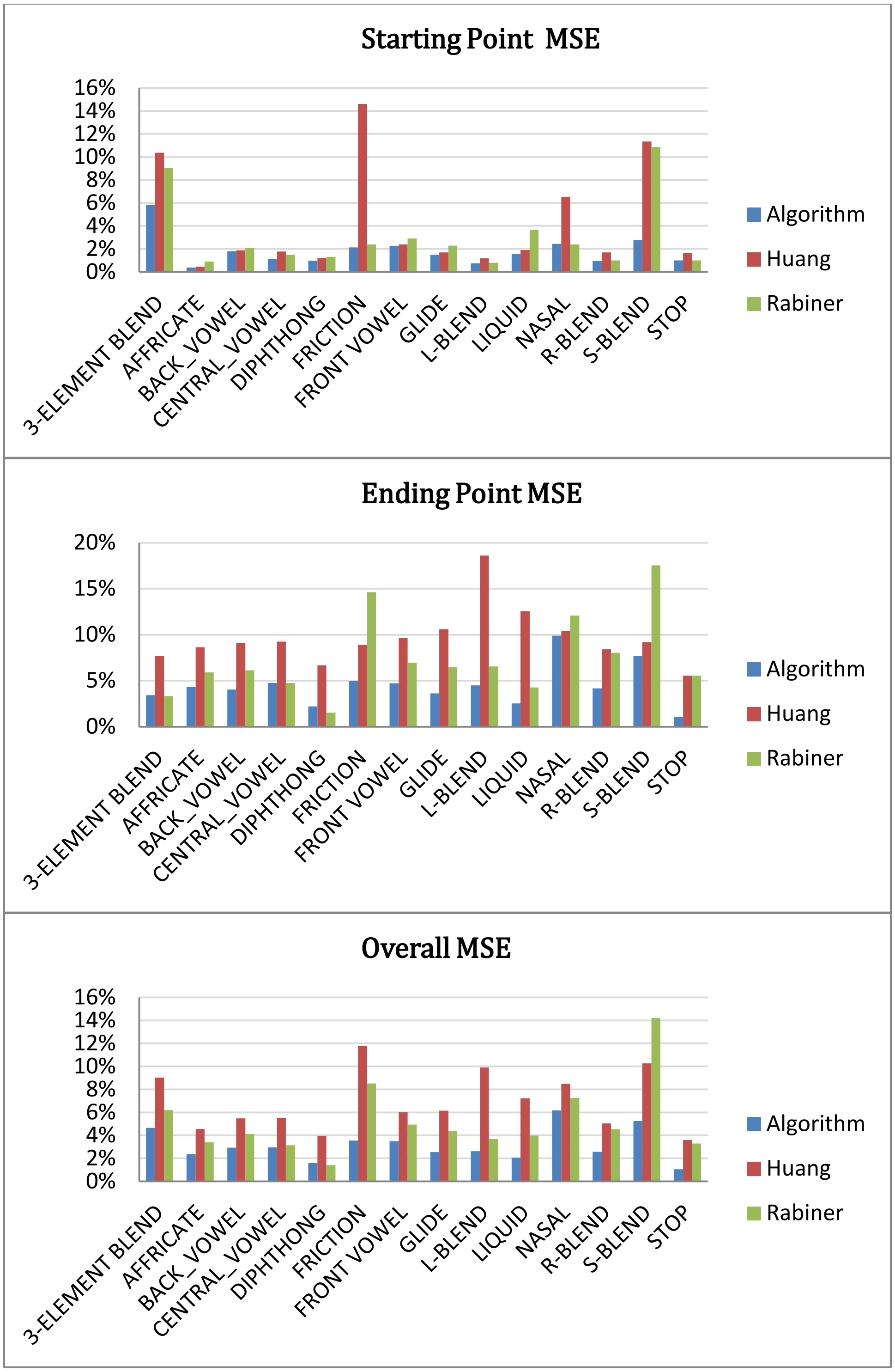


Figure 4.8: MSE per phoneme type, in case of pink noise (0dB≤ SNR<10dB).

Table 4.7: MSE per phoneme type, in case of pink noise (10dB ≤SNR<20dB).

Phoneme Type	Method	Mean Square Error		
		Starting Point	Ending Point	Overall
3-ELEMENT BLEND	Algorithm	3.3%	2.1%	2.7%
	Huang	4.5%	6.2%	5.4%
	Rabiner	8.3%	4.7%	6.5%
AFFRICATE	Algorithm	0.6%	1.3%	1.0%
	Huang	0.7%	3.8%	2.2%
	Rabiner	0.7%	3.4%	2.1%
BACK VOWEL	Algorithm	0.5%	2.5%	1.5%
	Huang	0.9%	6.8%	3.9%
	Rabiner	0.7%	5.7%	3.2%
CENTRAL VOWEL	Algorithm	0.5%	2.5%	1.5%
	Huang	0.7%	6.8%	3.8%
	Rabiner	0.7%	5.8%	3.2%
DIPHTHONG	Algorithm	**0.8%**	0.9%	0.8%
	Huang	0.9%	3.0%	1.9%
	Rabiner	0.5%	1.7%	1.1%
FRICTION	Algorithm	1.4%	2.5%	1.9%
	Huang	1.4%	10.1%	5.8%
	Rabiner	1.6%	7.9%	4.8%
FRONT VOWEL	Algorithm	1.1%	1.8%	1.5%
	Huang	1.1%	6.4%	3.7%
	Rabiner	1.3%	6.6%	4.0%
GLIDE	Algorithm	0.3%	2.0%	1.2%
	Huang	0.4%	6.5%	3.5%
	Rabiner	0.5%	5.6%	3.0%
L-BLEND	Algorithm	0.5%	1.5%	1.0%
	Huang	1.1%	7.2%	4.1%
	Rabiner	0.7%	3.7%	2.2%
LIQUID	Algorithm	0.3%	1.8%	1.0%
	Huang	0.4%	6.1%	3.3%
	Rabiner	0.4%	4.9%	2.6%
NASAL	Algorithm	0.4%	5.8%	3.1%
	Huang	0.4%	11.6%	6.0%
	Rabiner	0.9%	9.5%	5.2%
R-BLEND	Algorithm	2.0%	2.2%	2.1%
	Huang	2.5%	8.1%	5.3%
	Rabiner	2.2%	6.3%	4.3%
S-BLEND	Algorithm	3.7%	1.7%	2.7%
	Huang	4.0%	6.7%	5.4%
	Rabiner	11.1%	8.5%	9.8%
STOP	Algorithm	**1.2%**	0.8%	1.0%
	Huang	0.9%	6.1%	3.5%
	Rabiner	0.7%	5.6%	3.2%

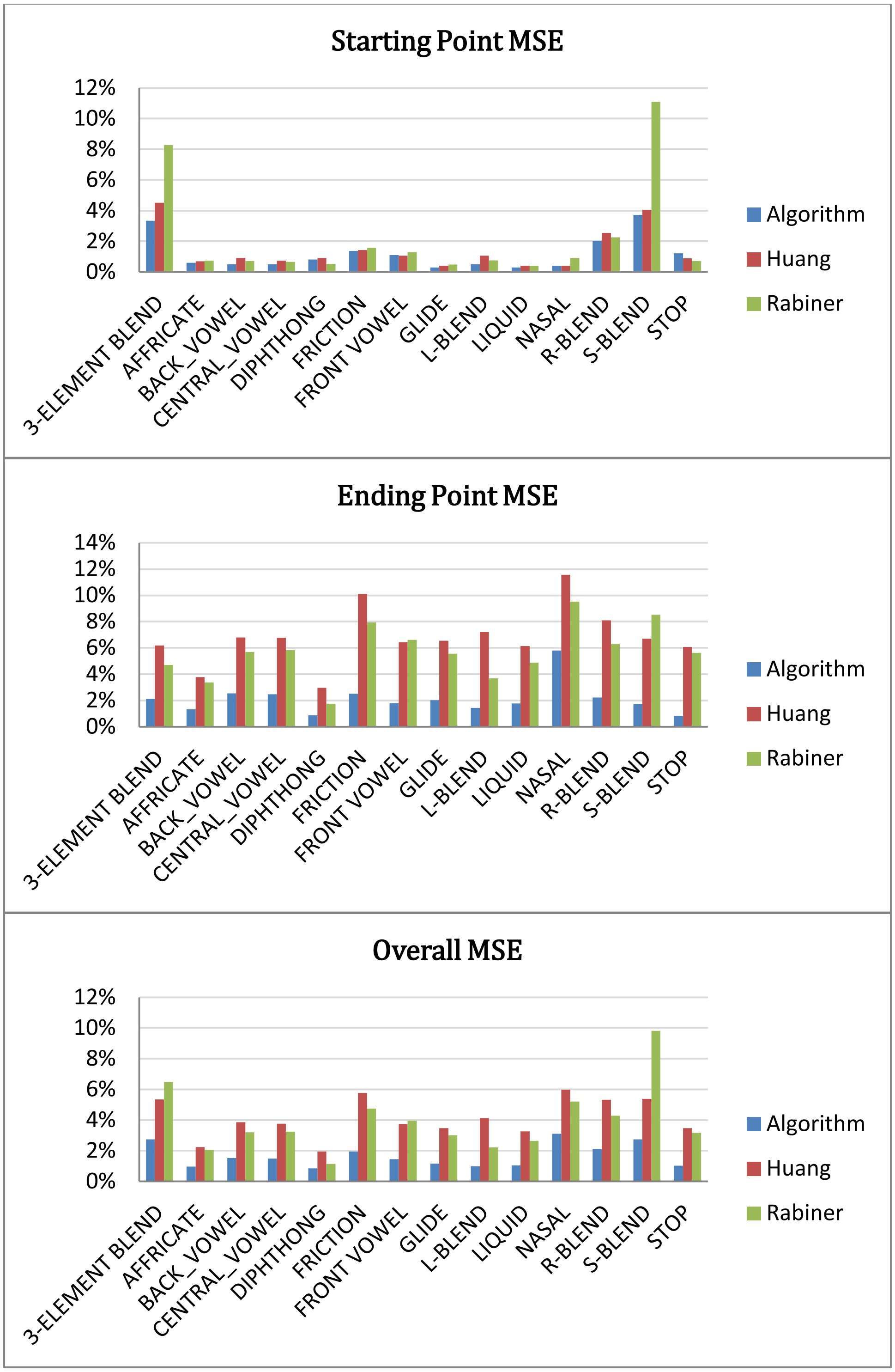


Figure 4.9: MSE per phoneme type, in case of pink noise (10dB≤ SNR<20dB).

Table 4.8: MSE per phoneme type, in case of pink noise (20dB ≤SNR≤30dB).

Phoneme Type	Method	Mean Square Error		
		Starting Point	Ending Point	Overall
3-ELEMENT BLEND	Algorithm	0.3%	0.8%	0.5%
	Huang	0.7%	5.3%	3.0%
	Rabiner	7.7%	4.6%	6.1%
AFFRICATE	Algorithm	0.6%	0.9%	0.7%
	Huang	0.7%	1.6%	1.1%
	Rabiner	0.8%	1.4%	1.1%
BACK VOWEL	Algorithm	0.9%	0.8%	0.8%
	Huang	0.9%	1.4%	1.1%
	Rabiner	0.9%	2.7%	1.8%
CENTRAL VOWEL	Algorithm	0.4%	0.9%	0.6%
	Huang	0.5%	3.3%	1.9%
	Rabiner	0.4%	3.8%	2.1%
DIPHTHONG	Algorithm	0.5%	0.8%	0.6%
	Huang	0.6%	1.4%	1.0%
	Rabiner	0.6%	1.5%	1.0%
FRICTION	Algorithm	0.7%	0.9%	0.8%
	Huang	0.9%	5.1%	3.0%
	Rabiner	1.2%	6.7%	4.0%
FRONT VOWEL	Algorithm	0.3%	0.6%	0.4%
	Huang	0.2%	1.8%	1.0%
	Rabiner	1.0%	1.8%	1.4%
GLIDE	Algorithm	**1.3%**	0.6%	0.9%
	Huang	1.2%	2.9%	2.1%
	Rabiner	1.1%	3.4%	2.3%
L-BLEND	Algorithm	0.7%	0.3%	0.5%
	Huang	0.9%	3.0%	1.9%
	Rabiner	0.7%	3.3%	2.0%
LIQUID	Algorithm	0.3%	0.5%	0.4%
	Huang	0.4%	2.8%	1.6%
	Rabiner	0.5%	4.8%	2.7%
NASAL	Algorithm	0.6%	3.9%	2.2%
	Huang	1.3%	7.6%	4.5%
	Rabiner	0.7%	8.6%	4.6%
R-BLEND	Algorithm	1.9%	1.1%	1.5%
	Huang	2.2%	3.3%	2.8%
	Rabiner	2.5%	4.3%	3.4%
S-BLEND	Algorithm	0.8%	0.8%	0.8%
	Huang	1.3%	2.8%	2.1%
	Rabiner	8.0%	4.5%	6.2%
STOP	Algorithm	0.5%	0.2%	0.4%
	Huang	0.8%	3.4%	2.1%
	Rabiner	0.7%	3.9%	2.3%

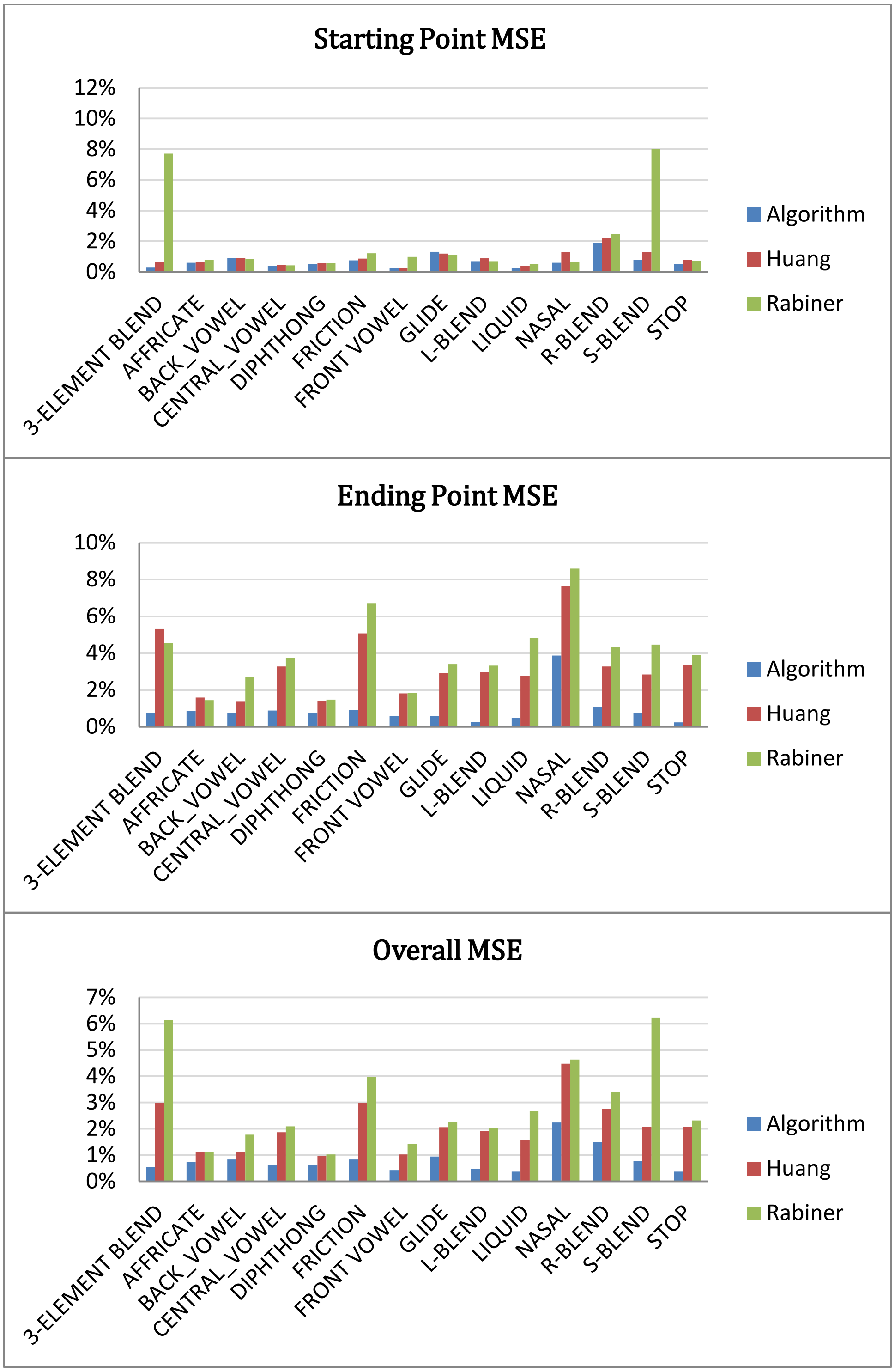


Figure 4.10: MSE per phoneme type, in case of pink noise (20dB ≤SNR≤30dB).

Table 4.9: MSE per phoneme type, in case of car noise (0dB ≤SNR<10dB).

Phoneme Type	Method	Mean Square Error		
		Starting Point	Ending Point	Overall
3-ELEMENT BLEND	Algorithm	3.5%	2.2%	2.9%
	Huang	3.9%	5.5%	4.7%
	Rabiner	4.3%	3.0%	3.6%
AFFRICATE	Algorithm	**2.7%**	1.8%	2.2%
	Huang	2.6%	2.6%	2.6%
	Rabiner	3.3%	1.9%	2.6%
BACK VOWEL	Algorithm	1.7%	1.5%	1.6%
	Huang	1.9%	7.1%	4.5%
	Rabiner	2.2%	1.6%	1.9%
CENTRAL VOWEL	Algorithm	1.5%	2.3%	1.9%
	Huang	1.6%	3.4%	2.5%
	Rabiner	1.9%	2.5%	2.2%
DIPHTHONG	Algorithm	0.5%	**1.2%**	0.9%
	Huang	0.9%	2.2%	1.5%
	Rabiner	0.7%	0.9%	0.8%
FRICTION	Algorithm	**1.0%**	2.0%	1.5%
	Huang	0.8%	5.6%	3.2%
	Rabiner	1.0%	3.0%	2.0%
FRONT VOWEL	Algorithm	0.6%	1.3%	1.0%
	Huang	1.1%	2.8%	1.9%
	Rabiner	0.9%	1.5%	1.2%
GLIDE	Algorithm	1.2%	1.3%	1.3%
	Huang	1.7%	4.2%	3.0%
	Rabiner	1.2%	1.8%	1.5%
L-BLEND	Algorithm	1.6%	1.5%	1.5%
	Huang	1.6%	4.7%	3.2%
	Rabiner	1.9%	2.1%	2.0%
LIQUID	Algorithm	0.5%	2.7%	1.6%
	Huang	0.8%	7.5%	4.2%
	Rabiner	1.3%	3.4%	2.4%
NASAL	Algorithm	1.8%	5.0%	3.4%
	Huang	2.9%	9.4%	6.1%
	Rabiner	1.8%	7.1%	4.5%
R-BLEND	Algorithm	**2.5%**	2.8%	2.6%
	Huang	3.1%	5.0%	4.0%
	Rabiner	2.1%	3.0%	2.6%
S-BLEND	Algorithm	4.2%	1.9%	3.1%
	Huang	5.7%	10.9%	8.3%
	Rabiner	6.1%	7.5%	6.8%
STOP	Algorithm	0.5%	1.5%	1.0%
	Huang	0.6%	2.7%	1.6%
	Rabiner	0.5%	2.1%	1.3%

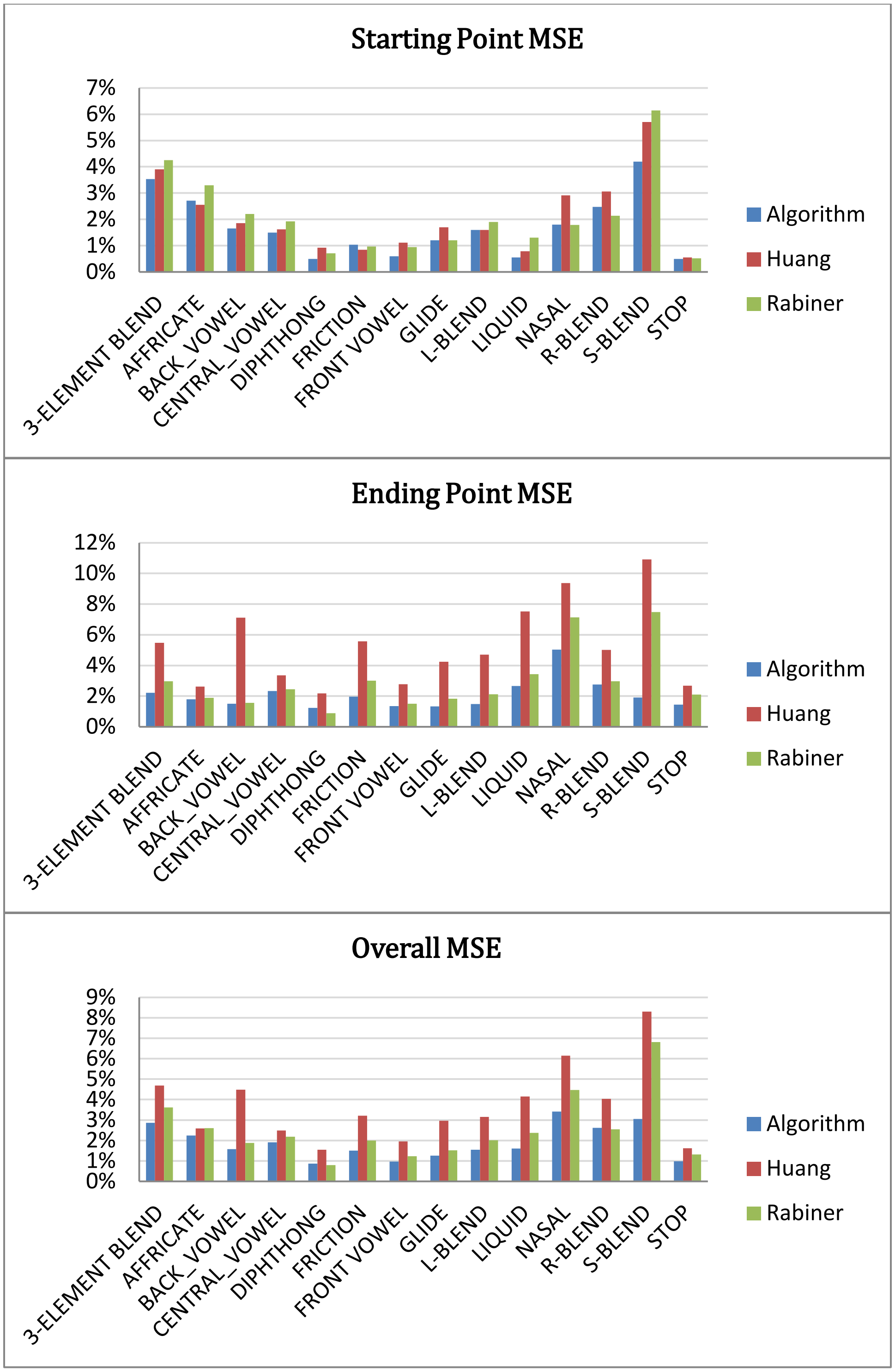
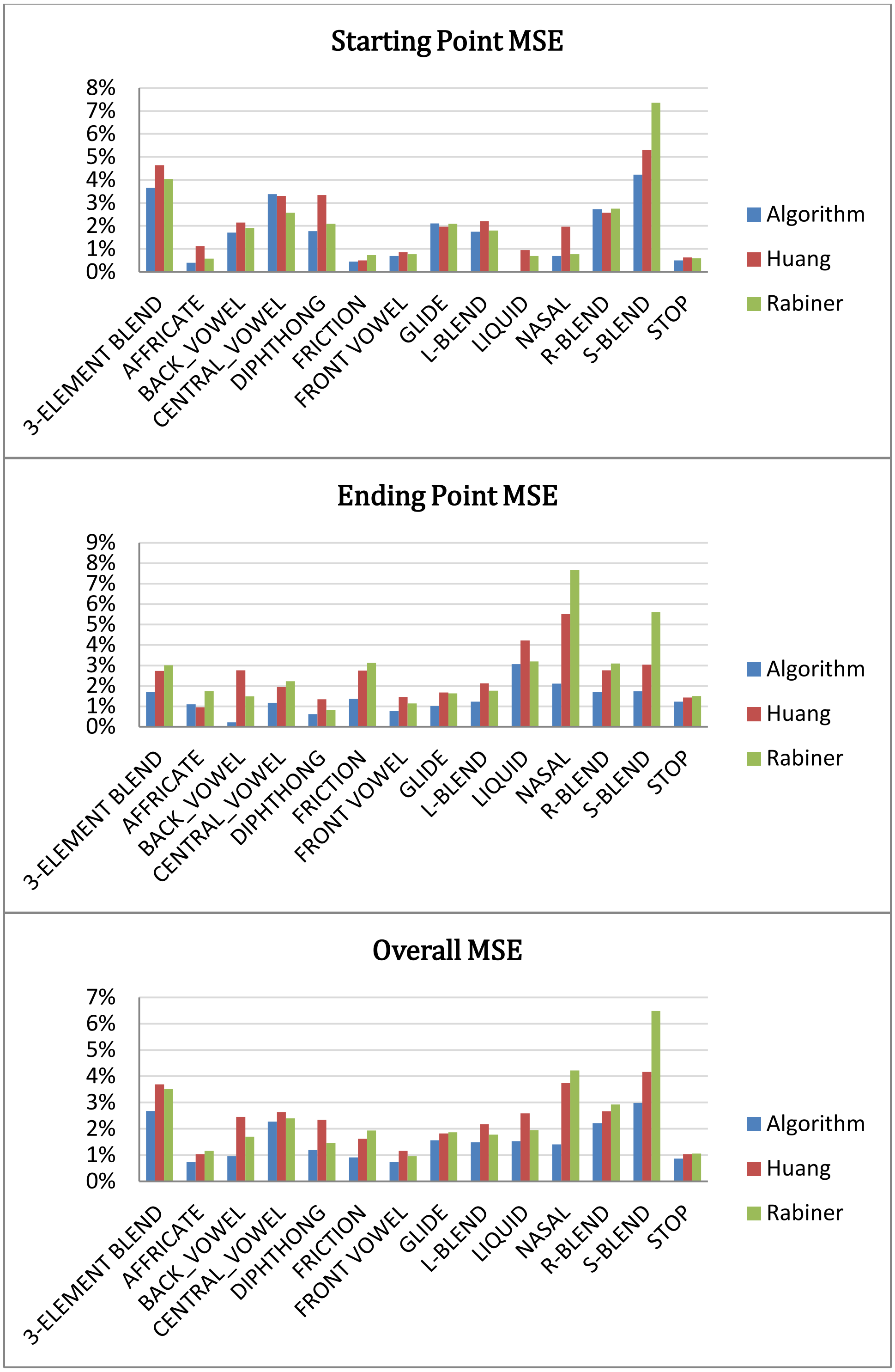


Figure 4.11: MSE per phoneme type, in case of car noise (0dB ≤SNR<10dB).

Table 4.10: MSE per phoneme type, in case of car noise (10dB≤ SNR<20dB).

Phoneme Type	Method	Mean Square Error		
		Starting Point	Ending Point	Overall
3-ELEMENT BLEND	Algorithm	3.7%	1.7%	2.7%
	Huang	4.6%	2.7%	3.7%
	Rabiner	4.0%	3.0%	3.5%
AFFRICATE	Algorithm	0.4%	**1.1%**	0.7%
	Huang	1.1%	1.0%	1.0%
	Rabiner	0.6%	1.8%	1.2%
BACK VOWEL	Algorithm	1.7%	0.2%	1.0%
	Huang	2.1%	2.8%	2.5%
	Rabiner	1.9%	1.5%	1.7%
CENTRAL VOWEL	Algorithm	**3.4%**	1.2%	2.3%
	Huang	3.3%	2.0%	2.6%
	Rabiner	2.6%	2.2%	2.4%
DIPHTHONG	Algorithm	1.8%	0.6%	1.2%
	Huang	3.3%	1.3%	2.3%
	Rabiner	2.1%	0.8%	1.5%
FRICTION	Algorithm	0.5%	1.4%	0.9%
	Huang	0.5%	2.7%	1.6%
	Rabiner	0.7%	3.1%	1.9%
FRONT VOWEL	Algorithm	0.7%	0.8%	0.7%
	Huang	0.9%	1.5%	1.2%
	Rabiner	0.8%	1.1%	1.0%
GLIDE	Algorithm	**2.1%**	1.0%	1.6%
	Huang	2.0%	1.7%	1.8%
	Rabiner	2.1%	1.6%	1.9%
L-BLEND	Algorithm	1.7%	1.2%	1.5%
	Huang	2.2%	2.1%	2.2%
	Rabiner	1.8%	1.8%	1.8%
LIQUID	Algorithm	0.0%	3.1%	1.5%
	Huang	1.0%	4.2%	2.6%
	Rabiner	0.7%	3.2%	1.9%
NASAL	Algorithm	0.7%	2.1%	1.4%
	Huang	2.0%	5.5%	3.7%
	Rabiner	0.8%	7.7%	4.2%
R-BLEND	Algorithm	**2.7%**	1.7%	2.2%
	Huang	2.6%	2.8%	2.7%
	Rabiner	2.8%	3.1%	2.9%
S-BLEND	Algorithm	4.2%	1.7%	3.0%
	Huang	5.3%	3.0%	4.2%
	Rabiner	7.4%	5.6%	6.5%
STOP	Algorithm	0.5%	1.2%	0.9%
	Huang	0.6%	1.4%	1.0%
	Rabiner	0.6%	1.5%	1.1%


Starting Point MSE

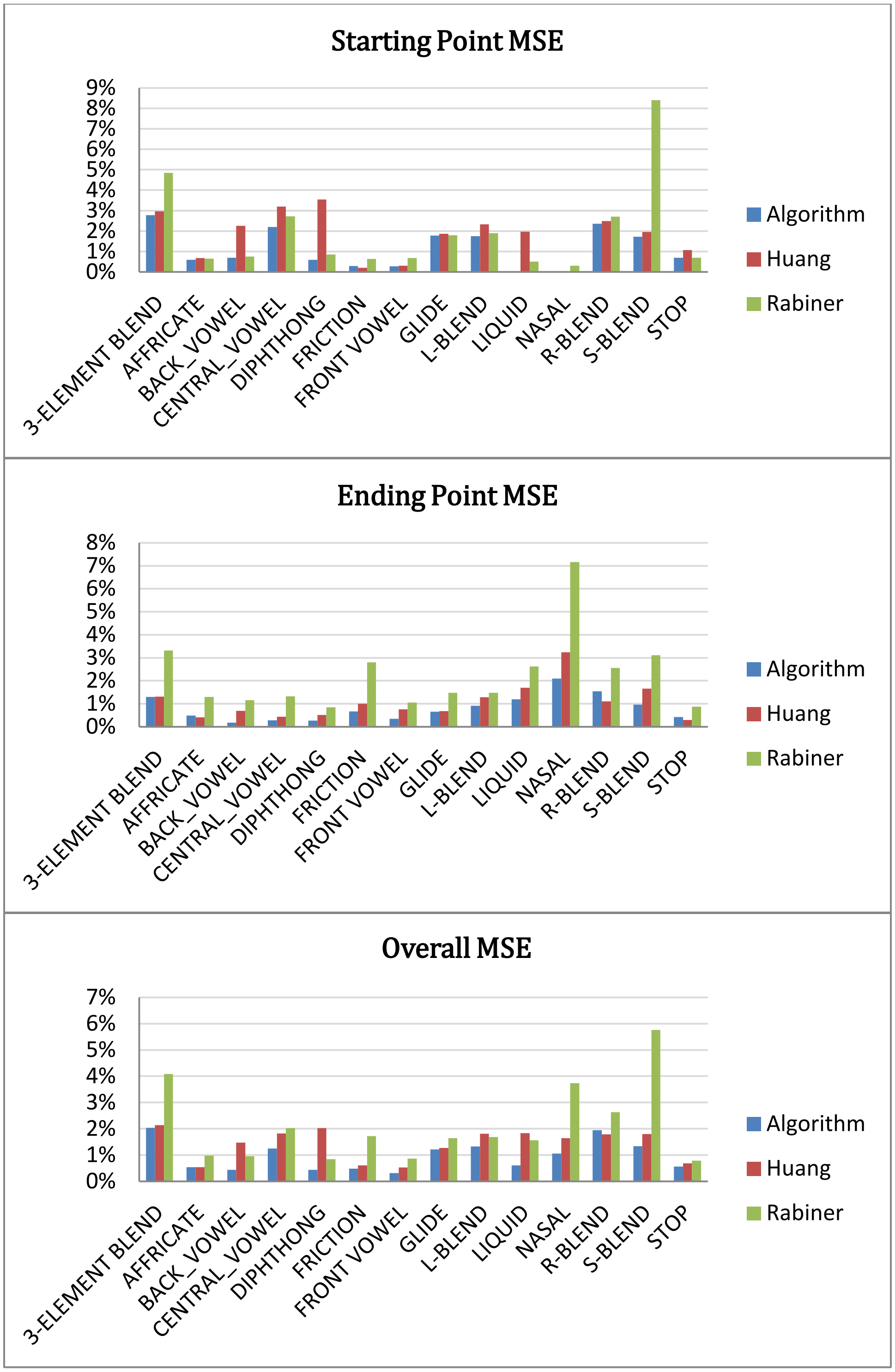
Ending Point MSE

Overall MSE

Figure 4.12: MSE per phoneme type, in case of car noise (10dB ≤SNR<20dB).

Table 4.11: MSE per phoneme type, in case of car noise (20dB≤ SNR<30dB).

Phoneme Type	Method	Mean Square Error		
		Starting Point	Ending Point	Overall
3-ELEMENT BLEND	Algorithm	2.8%	1.3%	2.0%
	Huang	3.0%	1.3%	2.1%
	Rabiner	4.8%	3.3%	4.1%
AFFRICATE	Algorithm	0.6%	**0.5%**	0.5%
	Huang	0.7%	0.4%	0.5%
	Rabiner	0.7%	1.3%	1.0%
BACK VOWEL	Algorithm	0.7%	0.2%	0.4%
	Huang	2.3%	0.7%	1.5%
	Rabiner	0.8%	1.2%	1.0%
CENTRAL VOWEL	Algorithm	2.2%	0.3%	1.2%
	Huang	3.2%	0.4%	1.8%
	Rabiner	2.7%	1.3%	2.0%
DIPHTHONG	Algorithm	0.6%	0.3%	0.4%
	Huang	3.5%	0.5%	2.0%
	Rabiner	0.8%	0.8%	0.8%
FRICTION	Algorithm	**0.3%**	0.7%	0.5%
	Huang	0.2%	1.0%	0.6%
	Rabiner	0.6%	2.8%	1.7%
FRONT VOWEL	Algorithm	0.3%	0.4%	0.3%
	Huang	0.3%	0.8%	0.5%
	Rabiner	0.7%	1.1%	0.9%
GLIDE	Algorithm	1.8%	0.7%	1.2%
	Huang	1.9%	0.7%	1.3%
	Rabiner	1.8%	1.5%	1.6%
L-BLEND	Algorithm	1.8%	0.9%	1.3%
	Huang	2.3%	1.3%	1.8%
	Rabiner	1.9%	1.5%	1.7%
LIQUID	Algorithm	0.0%	1.2%	0.6%
	Huang	2.0%	1.7%	1.8%
	Rabiner	0.5%	2.6%	1.6%
NASAL	Algorithm	0.0%	2.1%	1.1%
	Huang	0.0%	3.2%	1.6%
	Rabiner	0.3%	7.2%	3.7%
R-BLEND	Algorithm	2.4%	**1.5%**	**2.0%**
	Huang	2.5%	1.1%	1.8%
	Rabiner	2.7%	2.6%	2.6%
S-BLEND	Algorithm	1.7%	1.0%	1.3%
	Huang	2.0%	1.7%	1.8%
	Rabiner	8.4%	3.1%	5.8%
STOP	Algorithm	0.7%	**0.4%**	0.6%
	Huang	1.1%	0.3%	0.7%
	Rabiner	0.7%	0.9%	0.8%


Starting Point MSE

Ending Point MSE

Overall MSE

Figure 4.13: MSE per phoneme type, in case of car noise (20dB ≤SNR≤30dB).

4.3 Summary

This chapter presents the experimental work conducted to analyze the performance of the proposed endpoint detection algorithm. First, the overall performance has been compared to the reference algorithms in noise-free environment and in the presence of artificial noise contamination. In addition, extended comparisons have been investigated to evaluate the performance of the proposed algorithm and the reference algorithms as they applied to different types of American English Phonemes. In both sets of experiments, the proposed algorithm shows the best performance of all compared algorithm in most of the cases, as can be seen in the figures and tables provided in this chapter.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

The main contribution of this thesis is the introduction of a new method for speech endpoint detection. The proposed method presents a novel detection algorithm which has been derived from Chan-Vese algorithm [28] for image segmentation without edges. As opposed to edge-based image segmentation algorithms, Chan-Vese algorithm is based on solving an energy minimization problem, not on edges which are difficult to construct reliably in the presence of noise. In addition, our method has an ability to fuse numerous features of the speech signal, which allows one to find its endpoints in the presence of strong noises and uncertain transitions. In particular, Log-energy, PLP and MFCC features have been chosen along with multi-band spectral subtraction for enhancing the speech signal.

To quantitatively evaluate the performance of the proposed method, it was compared to two reference algorithms [5] and [6], which are discussed in details in Chapter 2. Extensive experiments involving different types of American English phonemes have demonstrated that the proposed method is more efficient than the reference algorithms. Even under conditions of severe noise contamination, the proposed method has also demonstrated reliable performance, as indicted by the metrics summarized in the provided tables and figures (see Chapter 4).

5.2 Future Work

This work shows that speech endpoint detection is a wide area that has been explored by many researchers over the past decades. However, practical issues remain challenging, and much progress still needs to be done to solve the problem completely.

Several paths to the continuation of this research can be outlined. First, further research might explore the performance of the proposed method using different techniques of speech enhancement such as nonlinear filters. Aside from conducting more research on the effect of speech enhancement techniques, another area to expand upon is investigating other types of features that would be more robust than those used in this thesis. Another possible direction of future work is to evaluate and compare the performance of the proposed algorithm in combination with an edge- based image segmentation method, e.g. [29]. In addition, the proposed algorithm is considered as an off-line technique, and thus is not yet suitable for real-time processing. It would be more useful if the proposed algorithm can be modified to support real-time applications, e.g. by using, [39]or [40].

References

[1] T. Martin, "Applications of limited vocabulary recognition systems," in *Rec.* 1974 *Symp. Speech Recognition,* D. R. Reddy, Ed. New York: Academic, 1975.

[2] L. F. Lamer, L. R. Rabiner, A. E. Rosenberg, and J. G. Wilpon, "An Improved Endpoint Detector for Isolated Word Recognition," *IEEE Transactions on Acoustics Speech and Signal Processing*, Vol. 29, No. 4, Aug. 1981.

[3] L. Rabiner and B. H. Juang, "*Fundamentals of Speech Recognition,*" Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1993.

[4] D. K. Freeman, G. Cosier, C. B. Southcott, and I. Boyd, "The Voice Activity Detector for The Pan-European Digital Cellular Mobile Telephone Service," In *Proc. ICASSP*, Vol.1, May 1989.

[5] L.R. Rabiner and M.R. Sambur, "An Algorithm for Determining the Endpoints of Isolated Utterances, "*Bell System Technical Journal*, Vol. 54, No. 2, 1975.

[6] L.S. Huang and C.H. Yang, "A Novel Approach to Robust Speech Endpoint Detection in Car Environments," in *Proc. ICASSP*, Vol.3, 2000.

[7] G. D. Wu and C. T. Lin, "Word Boundary Detection with Mel-Scale Frequency Bank in Noisy Environment," *IEEE Transactions on Speech and Audio Processing*, Vol. 8, No.5, Sep. 2000.

[8] J. R. Deller, J. H.L. Hansen and J. G. Proakis, "*Discrete-Time Processing of Speech Signals,"* The Institute of Electrical and Electronics Engineers, Inc., New York, 2000.

[9] L. Deng and D. O'Shaughnessy, "*Speech Processing: A Dynamic and Optimization-Oriented Approach,"* Marcel Dekker Inc.*,* New York, 2003.

[10] P. Loizou*, "Speech Enhancement: Theory and Practice*," Boca Raton, FL: CRC, 2007.

[11] G.S. Ying, C.D. Mitchell and L.H. Jamieson, "Endpoint Detection of Isolated Utterances Based on A Modified Teager Energy Measurement," *Proceedings of The IEEE International Conference on Acoustics Speech and Signal Processing,* Vol. 2, Apr. 1993.

[12] J. F. Kaiser, "On a Simple Algorithm to Calculate The 'Energy' of a Signal," *Proceedings of The IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol.1, Apr. 1990.

[13] J. L. Shen, J. W. Hung and L.S. Lee, "Robust Entropy-based Endpoint Detection for Speech Recognition in Noisy Environments," *International Conference on Spoken Language Processing*, Sydney, 1998.

[14] J. C. Junqua, B. Mak and B. Reaves, "A Robust Algorithm for Word Boundary Detection in the Presence of Noise," *IEEE Transactions on Speech and Audio Processing,* Vol.2*,* No.3, July 1994.

[15] B. F. Wu and K. C. Wang, "Robust Endpoint Detection Algorithm Based on the Adaptive Band-Partitioning Spectral Entropy in Adverse Environments," *IEEE Transactions on Speech and Audio Processing*, Vol. 13, No. 5, Sep. 2005.

[16] F. Yingle, L. Yi and W. Chuanyan, "Speech Endpoint Detection Based on Speech Time-Frequency Enhancement and Spectral Entropy," *Proceedings of the IEEE Engineering in Medicine and Biology 27th Annual Conference*, Sep. 2005.

[17] A. Lipeika and J. Lipeikiene, “Word Endpoint Detection Using Dynamic Programming,” *Informatica*, Vol. 14, No. 4, Sep. 2003.

[18] J. Sohn, N. S. Kim and W. Sung, “A Statistical Model-Based Voice Activity Detection,” *IEEE Signal Processing Letters*, Vol. 6, No. 1, Jan. 1999.

[19] M. Fujimoto, K. Ishizuka and H. Kato, “Noise Robust Voice Activity Detection Based on Statistical Model and Parallel Non-Linear Kalman Filter,” *ICASSP’07*, 2007.

[20] J. Wu and X. L. Zhang,” An Efficient Voice Activity Detection Algorithm by Combining Statistical Model and Energy Detection,” *EURASIP Journal on Advances in Signal Processing,* July 2011.

[21] H. Zhang and H. Hu , "An Endpoint Detection Algorithm Based on MFCC and Spectral Entropy Using BP NN," *2nd International Conference on Signal Processing Systems (ICSPS),* Vol.2, July 2010.

[22] Kh. Aghajani, M.T. Manzuri, M. Karami and H. Tayebi, "A Robust Voice Activity Detection Based on Wavelet Transform", *In Second International Conference on Electrical Engineering,* Mar. 2008.

[23] H. Ghaemmaghami, R. Vogt, S. Sridharan and M. Mason, “Speech Endpoint Detection Using Gradient Based Edge Detection Techniques,” *2nd International Conference on Signal Processing and Communication Systems. ICSPCS 2008*, Dec. 2008.

[24] J. Picone, “Signal Modeling Techniques in Speech Recognition,” *Proceedings of the IEEE, IEEE* 1993.

[25] H. Qiang and Z. Youwei, “On Prefiltering and endpoint detection of speech signal,” *Proceedings of ICSP '98, IEEE 1998*.

[26] H. Hermansky, “Perceptual Linear Predictive (PLP) Analysis of Speech,” *Journal of Acoustical Society of America*, Apr. 1990.

[27] S. Davis and P. Mermelstein, “Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 28, No. 4, Aug. 1980.

[28] T. Chan and L. Vese, “Active Contours without Edges,” *IEEE Transactions on Image Processing*, Vol. 10, No. 2, 2001.

[29] V. Caselles, F. Catté, T. Coll, and F. Dibos, “A Geometric Model for Active Contours in Image Processing,” *Numer. Math.*, Vol. 66, 1993.

[30] V. Caselles, R. Kimmel, and G. Sapiro, “On Geodesic Active Contours,” *Int. J. Comput. Vis.*, Vol. 22, No. 1, 1997.

[31] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active Contour Models,” *Int. J. Comput. Vis.*, Vol. 1, 1988.

[32] R. Malladi, J. A. Sethian, and B. C. Vemuri, “A Topology Independent Shape Modeling Scheme,” in *Proc. SPIE Conf. Geometric Methods Computer Vision II*, Vol. 2031, San Diego, CA, 1993.

[33] S. Osher and J. A. Sethian, “Fronts Propagating with Curvature Dependent Speed: Algorithms Based On Hamilton-Jacobi Formulations,” *Journal of Computational*

Physics, Vol. 79, No. 1, 1988.

[34] N. Zhang, J. Zhang; R. Shi , "An Improved Chan-Vese Model for Medical Image Segmentation," In 2008 *International Conference on Computer Science and Software Engineering,* , Vol. 1, 2008.

[35] Z. L. Szpak and J. R. Tapamo, "Maritime Surveillance: Tracking Ships Inside a Dynamic Background Using a Fast Level-Set", *Expert Systems with Applications: An International Journal,* Vol. 38, No. 6, June, 2011.

[36] A. J. Compton, Phonetic Symbols Table: Phonetic Symbols for the Constant and Vowel Sounds of American English, March 2010. [online] available at http://comptonpeslonline.com/phonetic_symbols_table.shtml

[37] Speech Segmentation Benchmark. Retrieved August 15, 2012, from http://speechsegmentbm.sourceforge.net/

[38] A. Varga, HJM Steeneken, M. Tomlinson, and D. Jones. The NOISEX-92 study on the effect of additive noise on automatic speech recognition. DRA Speech Research Unit, Malvern, England, Tech. Rep, 1992.

[39] Y. Shi and W. C. Karl, "A Fast Level Set Method without Solving PDEs," *IEEE International Conference on Acoustics, Speech, and Signal Processing,* 2005.

[40] Y. Shi and W. C. Karl, " A Real-Time Algorithm for The Approximation of Level-Set Based Curve Evolution," *IEEE Trans. Image Processing,* Vol. 17, No. 5, May 2008.

[41] M. Petrou and J. Kittler, "Optimal edge detectors for ramp edges," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, May 1991.

[42] E. Carlstein, M. Muller, and D. Siegmund, *Change-Point Problems* . Hayward, CA: Inst. Math. Statist., 1994.