Front-end noise reduction algorithms for automatic speech recognition

Dai, Peng

2014

Dai, P. (2014). Front-end noise reduction algorithms for automatic speech recognition.
Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/61677>

<https://doi.org/10.32657/10356/61677>

**FRONT-END NOISE REDUCTION
ALGORITHMS FOR AUTOMATIC SPEECH
RECOGNITION**

DAI PENG

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2014

*To My family,
for their encouragement and love.*

Acknowledgements

Firstly, I would like to express my most sincere gratitude to my supervisor, Associate Professor Soon Ing Yann, for all these years of conscientious supervision, sound advice, invaluable guidance, constant encouragement and support. He is an oasis of ideas and his passion in science has inspired me and enriched my training as a student and a researcher.

I am deeply grateful to my family for their support and unfailing blessings. My dad, Mr. Dai Jianwei, my mum, Ms. Zhang Nina and my wife, Ms. Teng Xue. Their endless love, support, encouragement and care are a strong motivation for me.

I gratefully acknowledge the Nanyang Technological University for its financial support during the entire period of my candidature. I would also like to thank all the members and staff of the Media Technology Lab for providing me with a comfortable and pleasant working environment for doing my research.

Last but not least, I would like to thank all those who have helped me in the past.

Summary

One of the biggest obstacles that hinders the widespread use of automatic speech recognition technology is the inability to handle noise, which includes environmental noise, channel distortion and speaker variability, etc. Towards this end, we propose several feature compensation approaches to improve the robustness of automatic speech recognition (ASR) systems: 1) direct implementation of masking effect; 2) 2D psychoacoustic filter; 3) model based noise reduction. The first two are based on psychoacoustics, and the last one includes several algorithms based on a novel feature model. More details are given as follows.

The human auditory system can work properly in adverse environments, e.g. in a crowded shopping mall where thousands of people are talking loudly together with the background commercial broadcast. Therefore, modeling the human auditory system is a straightforward and logical approach to improve the performance of ASR systems. The first part of this thesis focuses on the study of masking effects, which describes how a clearly audible sound (maskee) becomes less audible because of the presence of another sound (masker). Masking effects can be classified as temporal masking and frequency masking (a.k.a. simultaneous masking). Chapter 3 introduces a novel Mel-Frequency Cepstral Coefficients (MFCC) based algorithm which simulates the properties of the human auditory system. It sequentially implements temporal masking and frequency masking in the time domain and the frequency domain, respectively.

For the second contribution on psychoacoustics, we further investigate the special property of the time-frequency domain and propose the 2D psychoacoustic filter.

In the time-frequency domain, the speech signal is represented over both time and frequency, which provides us the chance to address another psychoacoustic problem, i.e. temporal frequency masking. Temporal frequency masking describes the situation where the masker and maskee possess both different frequency and different commencing time. The 2D psychoacoustic filter implements not only temporal masking and frequency masking, but also temporal frequency masking and temporal integration. We also propose a unified model for the 2D psychoacoustic filter, which effectively models the equivalent masking phenomena. Mathematical derivations are provided to show the correctness of the 2D psychoacoustic filter based on the characteristic functions of masking effects.

The degradation of ASR performance is mainly due to the mismatch between the statistical model trained from the clean speech and the test features derived from the noisy speech. To reduce the mismatch, we propose to recover the clean speech from the noisy speech. Two different front-end noise reduction algorithms are presented, i.e. Smoothing & Noise Subtraction (SNS) and Newton & Log Power Subtraction (NLPS). SNS tries to recover the temporal structure of the speech power spectrum. The histogram of average speech log power spectrum shows that the contamination of noise leads to a shift of the noise peak. A two-step scheme is proposed to remove noise by first reducing the noise variance and then shifting the noise mean. As for NLPS, it works by solving a nonlinear function derived from the MFCC feature extraction algorithm.

Contents

Acknowledgements	i
Summary	ii
List of Figures	viii
List of Tables	x
List of Abbreviations	xii
List of Symbols	xiv
1 Introduction	1
1.1 Motivation	1
1.2 Objectives	4
1.3 Contributions	4
1.4 Organization of this Thesis	5
2 Literature Review	7
2.1 General Description	7
2.2 Fundamentals of Speech Recognition	9
2.2.1 Feature Extraction	10
2.2.2 Hidden Markov Model	14
2.3 Frontend Noise Removal Algorithms	17
2.3.1 Spectral Filtering	19
2.3.2 Subspace Filtering	23

2.3.3	Stereo Data Based Noise Removal	25
2.3.4	Cepstral Filtering	27
3	Psychoacoustic Modeling	28
3.1	Introduction	28
3.2	Temporal Masking	30
3.2.1	Type I Forward Masking Modeling	31
3.2.2	Type II Forward Masking Modeling	34
3.3	Simultaneous Masking	34
3.4	Temporal Spectral Average (TSA)	37
3.5	Results and Discussion	37
3.5.1	Implementation Details	37
3.5.2	Experimental Results	38
3.6	Conclusion	46
4	2D Psychoacoustic Filtering	47
4.1	Introduction	47
4.2	2D Psychoacoustic Modeling	48
4.2.1	Temporal Masking	48
4.2.2	Simultaneous Masking	51
4.2.3	Overall Joint Masking Effects	52
4.2.4	Temporal Integration	55
4.3	Parameter Design	58
4.3.1	Original 2D Filter	58
4.3.2	Temporal Warped (TW) 2D Psychoacoustic Filter	60
4.3.3	Temporal Frequency Warped (TFW) 2D Psychoacoustic Filter	64
4.3.4	Adaptive 2D Psychoacoustic Filter	68
4.4	Theoretical Analysis	73
4.4.1	Complex Spectral Processing	73
4.4.2	Noise Removal	73
4.5	Results and Discussion	75
4.5.1	Implementation Details	75

4.5.2	Experimental Results	75
4.6	Conclusion	80
5	2D Psychoacoustic L-Filter	81
5.1	Introduction	81
5.2	Algorithm Description	83
5.2.1	Algorithm Description	83
5.2.2	Theoretical Analysis	86
5.3	2D L-filter design	88
5.4	Results and Discussion	89
5.4.1	Experimental Results	89
5.4.2	Results Analysis	91
5.5	Conclusion	93
6	Multiple Model Feature Compensation	95
6.1	Introduction	95
6.2	Smoothing & Noise Subtraction (SNS)	99
6.2.1	General Description	99
6.2.2	2D Smoothing	102
6.2.3	Noise Subtraction	104
6.3	Newton & Log Power Subtraction (NLPS)	107
6.3.1	General Description	107
6.3.2	Iterative Solution	108
6.3.3	Prior Estimates	110
6.3.4	Log-Power Subtraction (LPS)	111
6.4	Results and Discussion	113
6.4.1	Experimental Results	113
6.4.2	Results Analysis	116
6.5	Conclusion	118
7	Conclusion and Future Works	119
7.1	Conclusion	119

7.2 Further Works	121
Author's Publications	123
Bibliography	125

List of Figures

2.1	Speech recognition process.	9
2.2	Block Diagram of MFCC Feature Extraction Algorithm.	11
2.3	Mel-scaled triangular filter bank.	12
2.4	The Markov Generation Model (taken from HTKbook).	16
2.5	Different Domains in MFCC Feature Extraction Frontend.	18
3.1	Sketch of psychoacoustic data (temporal masking).	30
3.2	Sketch of psychoacoustic data (frequency masking).	35
3.3	Block Diagram of LTFC.	38
3.4	Experimental results.	41
3.5	Clean training condition results for babble noise.	45
4.1	Sketch of psychoacoustic data.	49
4.2	Joint Masking Effect.	53
4.3	TI experimental results [1].	56
4.4	1D Mexican Hat.	59
4.5	Warped Mexican Hat.	61
4.6	Psychoacoustic Data of FM.	65
4.7	Masking Effect.	66
4.8	Design of 2D Psychoacoustic Filter.	67
4.9	Block diagram of adaptive 2D psychoacoustic filtering.	70
4.10	Adaptive 2D Psychoacoustic Filtering.	70
4.11	Spectrogram of digit string '3Z82' from the AURORA2 database. . .	74
4.12	System Diagram of Adaptive 2D Psychoacoustic Filtering Algorithm. .	75
4.13	Experimental results (clean training condition).	79

4.14	Experimental results (multi training condition).	80
5.1	Example of equivalent masking.	82
5.2	Sample speech signal.	87
5.3	System Diagram of the Proposed Algorithm	91
6.1	Effect of Noise on Speech Power Spectrum.	97
6.2	Effect of SS on Speech Histogram.	98
6.3	Effect of Noise on Speech Power Spectrum.	100
6.4	Effect of Noise on Speech PDF.	100
6.5	Overlapping of Two PDFs.	101
6.6	Block diagram of SNS.	102
6.7	Example of the proposed algorithm.	103
6.8	Speech PDF: digit string ‘3Z82’ from the AURORA2 database.	105
6.9	Diagram of NLPS.	111
6.10	Relative Improvements for Clean Training Condition.	117

List of Tables

3.1	Forward Masking Model Parameters.	32
3.2	Centre Frequencies for Mel Scaled Filter Banks.	33
3.3	Recognition Results of LTFC for Clean Training Condition (%). . . .	39
3.4	Recognition Results of LTFC for Multi Training Condition(%). . . .	40
3.5	Recognition results for comparison targets under clean training condition (%).	41
3.6	Recognition results for comparison targets under multi training condition (%).	42
3.7	Relative Improvements (%).	43
3.8	Relative Improvements (%).	44
4.1	Temporal Masking Parameters [2].	49
4.2	Initial Parameter Set of 2D Filter.	59
4.3	The Original 2D Filter.	60
4.4	Initial Parameter Set of the Temporal Warped 2D Filter.	63
4.5	Temporal Warped 2D Psychoacoustic Filter.	64
4.6	Design of LI Parameters.	65
4.7	Temporal Frequency Warped 2D Psychoacoustic Filter.	68
4.8	Temporal Integration Parameter.	69
4.9	Temporal Frequency Warped 2D Psychoacoustic Filter (low band). .	71
4.10	Temporal Frequency Warped 2D Psychoacoustic Filter (high band). .	72
4.11	Recognition Results of Adaptive 2D Filter (%) for Clean Training Condition.	76
4.12	Recognition Results of Adaptive 2D Filter (%) for Multi Training Condition.	76

4.13	Recognition results for comparison targets under clean training condition (%)	77
4.14	Recognition results for comparison targets under multi training condition (%)	77
4.15	Relative Improvements under clean training condition (%)	78
4.16	Relative Improvements under multi training condition (%)	78
5.1	Adaptive 2D Psychoacoustic L-Filter: low band.	89
5.2	Adaptive 2D Psychoacoustic L-Filter: high band.	90
5.3	Recognition Results of Adaptive 2D L-Filter (%) for Clean Training Condition.	91
5.4	Recognition Results of Adaptive 2D L-Filter (%) for Multi Training Condition.	92
5.5	Recognition results for comparison targets (%)	92
5.6	Comparison of H-filter and L-filter (Avg 0-20).	93
6.1	Recognition results for comparison targets under multi training condition (%)	105
6.2	Recognition Results of SNS (%) for Clean Training Condition	113
6.3	Recognition Results of SNS (%) for Multi Training Condition	113
6.4	Experimental Results for Different Parts of SNS	114
6.5	Experimental Results for Different Iterations	114
6.6	Recognition Results of NLPS (%) for Clean Training Condition	114
6.7	Recognition Results of NLPS (%) for Multi Training Condition	115
6.8	Recognition Results for Comparison Targets (%)	115
6.9	Relative Improvements for Clean Training Condition (%)	116
6.10	Relative Improvements for Multi Training Condition (%)	117

List of Abbreviations

ABBREVIATIONS	FULL EXPRESSIONS
1D	One Dimensional
2D	Two Dimensional
2DFT	2D Fourier Transform Filtering
ANN	Artificial Neural Network
AR	Autoregressive
ARMA	Autoregressive Moving Average
ASR	Automatic Speech Recognition
BM	Backward Masking
CMVN	Cepstral Mean and Variance Normalization
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DTW	Dynamic Time Warping
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FM	Forward Masking
HMM	Hidden Markov Model
IIR	Infinite Impulse Response
LI	Lateral Inhibition
LLR	Log-Likelihood Ratio
LPC	Linear Predictive Coding
LTFC	LI, TI, FM & CMVN
MFCC	Mel Frequency Cepstral Coefficients
MMSE	Minimum Mean Square Error
MSE	Mean Square Error
MVA	Mean Variance Normalization and ARMA filtering
NLPS	Newton and Log Power Subtraction
PLP	Perceptual Linear Predictive
RASTA	Relative Spectra

SegSNR	Segmental Signal to Noise Ratio
SNR	Signal to Noise Ratio
SNS	Smoothing and Noise Subtraction
SPLICE	Stereo-based Piecewise Linear Compensation for Environments
SS	Spectral Subtraction
STFT	Short-time Fourier Transform
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TFW	Temporal Frequency Warped
TI	Temporal Integration
TM	Temporal Masking
TW	Temporal Warped
VAD	Voice Activity Detection
WF	Wiener Filtering

List of Symbols

SYMBOL	MEANING
$\overline{(\cdot)}$	Mean function
α	Parameter in decision-directed approach
$\alpha(f, t)$	Adaptive parameter α
α_{TI}^{low}	Temporal integration parameter for low band speech
α_{TI}^{high}	Temporal integration parameter for high band speech
$\varepsilon(\omega)$	A small constant to guarantee positive of SS processed speech
$\gamma(f, t)$	The a-posteriori signal to noise ratio at frequency f and time t
θ_Y	Phase information of noisy speech signal
$\lambda_{NN}(\omega)$	Expectation of noise signal power spectrum
$\lambda_s(k)$	k th eigenvalue of the autocorrelation of a speech signal
$\lambda_{SS}(\omega)$	Expectation of clean speech power spectrum
$\lambda_{SY}(\omega)$	Cross-power spectrum of clean speech and noisy speech
$\lambda_{YY}(\omega)$	Expectation of noisy speech power spectrum
Λ_s	Eigenvalue matrix of the autocorrelation of a speech signal
Λ_μ	Diagonal matrix of Lagrange multipliers
$\mu(k)$	k th Lagrange multiplier

$\nu(m)$	A gain function
$\xi(f, t)$	The a-priori signal to noise ratio (SNR) at frequency f and time t
$\hat{\xi}(f, t)$	The a-priori SNR estimated by decision-directed approach
σ_n^2	Noise Variance
σ	The variance of Gaussian distribution
$\text{diag}(\cdot)$	Diagonal matrix operator
$a(\cdot)$	Temporal masking parameter
$b(\cdot)$	Temporal masking parameter
$c(\cdot)$	Temporal masking parameter
e	Base of natural logarithm
$E(\cdot)$	Expectation Function
f_s	Sampling frequency
$g(t)$	Traditional noise reduction Filter in time domain
G_1	Full rank $M \times M$ matrix
G	$= \begin{bmatrix} G_1 & 0 \\ 0 & 0 \end{bmatrix}$
$G(f)$	Spectral expression of traditional noise reduction filter $g(t)$
γ_{snr}	Signal to noise ratio
H_0	Hypothesis of speech absence
H_1	Hypothesis of speech presence
$H(f)$	A low-pass finite impulse response (FIR) filter
H_{sub}	Linear filter of subspace speech enhancement
I_0	Zero order modified Bessel function
J_α	Error between estimated a-priori SNR $\hat{\xi}(m, k)$ and real one $\xi(m, k)$

k	Sample index
$\max(\cdot)$	Maximum operator
$M(; ;)$	Confluent hypergeometric function
M_n	Average Magnitude Function at time n
$M_{fm}(\cdot)$	The amount of forward masking
$M_{bm}(\cdot)$	The amount of backward masking
$M_{li}(\cdot)$	The amount of simultaneous masking
$Mask_{-1}$	2D psychoacoustic L-filter
$n(k)$	Noise signal at the k -th sample
$N(f, t)$	Time frequency labeled coefficient of noise signal
$N(f_i, t_i)$	Time frequency index labeled coefficient of noise signal
$R(n)$	Autocorrelation function
$\text{Re}(\cdot)$	Real-part operator
R_s	Autocorrelation matrix of clean speech signal
t_i	Frame index
U	Orthogonal eigenvector matrix of a speech signal
$w(\cdot)$	Window function
$x(k)$	Clean speech signal
$X(f, t)$	Time frequency labeled coefficient of clean speech signal
$X(f_i, t_i)$	Time frequency index labeled coefficient of clean speech signal
$y(k)$	Noisy speech signal
$Y(f, t)$	Time frequency labeled coefficient of noisy speech signal
$Y(f_i, t_i)$	Time frequency index labeled coefficient of noisy speech signal

Chapter 1

Introduction

1.1 Motivation

Speech processing technology has drawn significant attention due to its numerous applications, e.g. hearing aids, cellular telephone, automated dialogue system, and real-time spoken language translation [3–12]. Speech processing technology consists of many different areas, such as speech enhancement, speech recognition and so forth [3, 5, 13–15]. In this thesis, we will investigate the noise robustness issues in automatic speech recognition (ASR) systems and propose different algorithms to improve the performance of ASR systems in adverse environments.

After decades of development, state-of-the-art speech recognition algorithms can work very well with clean speech. The word recognition accuracy can reach over 99% for small vocabulary tasks and over 90% for large vocabulary tasks [3, 11, 12, 16, 17]. However, with noise added in, the performance of ASR systems falls dramatically. Therefore, the development of noise-robust speech recognition algorithms is of vital importance [5, 18, 19]. Due to the imperfections in speech recording, trans-

mission, and storage, speech signal is often severely degraded. The contamination of the speech signal degrades the performance of automatic speech recognition systems [20–26]. An ASR system can be generally divided into two parts, i.e. feature extraction and pattern matching. In the feature extraction part, speech signals are transformed into feature vectors based on certain feature extraction algorithms, such as Mel-Frequency Cepstral Coefficient (MFCC) and Perceptual Linear Predictive (PLP) coefficients. In the pattern matching part, ASR systems usually model the speech using hidden Markov model (HMM) and adopt the maximum a posteriori (MAP) decision rule for classification [4, 5, 7, 27–29]. If the testing material matches the training material well, good recognition accuracies can be obtained. On the other hand, if the testing material and training material show different statistical properties due to certain reasons, e.g. environmental noise, channel distortion, etc, the recognition accuracies will become very poor. The degradation of ASR performance is mainly due to the mismatch between the statistical model trained from the clean speech and the test features derived from the noisy speech [23–25, 30].

Corresponding to the above mentioned two parts of ASR systems, there are generally two different categories of algorithms to solve the feature/model mismatch problem, namely, the feature approach and the model approach. The former works in the feature extraction part. It aims to remove the noise and recover the clean speech from the noisy speech. The algorithms described in this thesis belong to the feature approach. The model approach works in the pattern matching part, where the speech models are modified to match the statistical properties of the test speech. Examples of this approach include Support Vector Machine (SVM) / Hidden Markov Model (HMM) hybrid method [31], maximum likelihood linear regression adaptation (MLLR) [32], soft margin feature extraction (SMFE) [33], ensemble modeling [19, 34], STAtistical Reestimation (STAR) technique [35], etc.

For the feature approach, speech enhancement algorithms are commonly adopted for noise removal. Traditional solution to the noise problem is to filter the noisy speech using an additional black box. Some more advanced systems combine the speech enhancement algorithm within the feature extraction part. Much work has been done by using traditional signal processing techniques, such as spectral subtraction, Wiener filtering, and Minimum Mean Square Error (MMSE) [36–44]. Such approaches are very successful because the reconstruction of clean speech spectra is nearly optimal under the given assumptions. Recently, a number of algorithms which require the so called stereo data have become popular due to the promising ability to recover clean speech. Stereo data contains both the clean and noisy version of the same speech sample. Usually, front-end noise removing algorithms require both the training and testing materials to be processed by the proposed algorithm to make up for the distortion. However, the stereo data based algorithms can work merely on the HMM models trained from the original clean speech [16, 45]. The main idea of the above mentioned algorithms is to estimate the clean speech from the noisy speech based on certain statistical models.

There is also another kind of approach which uses simple smoothing methods, which manages to achieve very promising results. One of the most significant differences between speech enhancement and speech recognition is that speech recognition does not necessarily need to estimate the ‘clean’ speech. It is because robust speech recognition algorithms are designed for computers but not for human listening. Mean subtraction Variance normalization & ARMA filtering (MVA), Temporal Structure Normalization (TSN) and the 2D psychoacoustic filter are good examples [46–50]. They simply normalize or smooth the temporal structure of the speech. Their theoretical basis lies in that the temporal differences between clean and noisy features can be reduced by smoothing or a specially designed normaliza-

tion filter, which in return leads to satisfactory results. The speech processed by the above mentioned algorithms is different from the original clean speech. However, the computer can recognize it easily.

1.2 Objectives

In this thesis, different kinds of noise reduction algorithms are studied. The main objectives of this thesis are:

- to study and apply psychoacoustic models to ASR systems.
- to develop a novel 2D psychoacoustic model which can implement Temporal Masking, Simultaneous Masking, and Temporal Integration at the same time.
- to derive new noise removal algorithms based on the MFCC feature extraction scheme.

1.3 Contributions

The major contributions of this dissertation are summarized as follows:

- A new auditory modeling algorithm is proposed. The algorithm successfully implements three different kinds of psychoacoustic effects, Temporal Masking, Simultaneous Masking, and Temporal Integration in a sequential manner.
- A novel hybrid feature extraction algorithm is proposed based on the threshold model of psychoacoustic effects. It implements Forward Masking, Lateral Inhibition, and Temporal Integration with a simple high pass 2D psychoacoustic filter. Detailed mathematical derivation is provided to show the correctness of

the 2D psychoacoustic filter based on the characteristic functions of masking effects.

- The low-pass 2D psychoacoustic filter (L-filter) is derived based on the adaptation model of masking effects. The new 2D filter manages to fully explore Temporal Integration effects and yields good results.
- Two different kinds of front-end noise reduction algorithms are presented, i.e. Smoothing & Noise Subtraction (SNS) as well as Newton & Log Power Subtraction (NLPS). They try to recover the clean speech based on certain statistical model.
- Comparison of all the proposed techniques and some state-of-the-art speech recognition algorithms is performed.

1.4 Organization of this Thesis

This thesis consists of seven chapters. In Chapter 2, a brief summary of speech recognition algorithms is presented so as to pave the way to the original works in the following chapters.

In Chapter 3, detailed discussion of different psychoacoustic models is presented. Discussion is focused on masking effects, including temporal masking and simultaneous masking. A novel hybrid front-end noise removal algorithm is proposed, which successfully implements different kinds of masking effects sequentially.

In Chapter 4, the proposed 2D psychoacoustic filter is elaborated at length. The proposed algorithm implements three different kinds of psychoacoustic effects (temporal masking, lateral inhibition, and temporal integration) with a novel 2D psychoacoustic filter.

In Chapter 5, a unified system model is proposed to explain the equivalent masking phenomena. We successfully explain different psychoacoustic phenomena in a unified framework. Detailed theoretical analysis is provided to show the advantage of the 2D psychoacoustic filter.

In Chapter 6, two different kinds of front-end noise removal algorithms are presented, i.e. Smoothing & Noise Subtraction (SNS) and Newton & Log Power Subtraction (NLPS). SNS tries to recover the temporal structure of speech power spectrum. The histogram of average speech log power spectrum shows that the contamination of noise leads to a shift of the noise peak, which in return degrades the performance of speech recognition systems. A two-step scheme is proposed to weaken the noise effects by first reducing the noise variance and then shifting the noise mean. As for NLPS, it works by direct solution of the nonlinear function derived from the MFCC feature extraction algorithm.

Finally, Chapter 7 summarizes this thesis and discusses possible future research directions.

Chapter 2

Literature Review

2.1 General Description

In this chapter, we will first give a general description of a typical HMM based ASR system, and then briefly review several front-end noise removal algorithms. Speech recognition is the technology that can convert speech (spoken words captured by a microphone or other devices) into text. The first serious speech recognizer called Audrey was developed in 1952 by Davis, Biddulph and Balashek of Bell Labs. The system was able to achieve 97% accuracy on the spoken forms of ten digits [30]. However, the method adopted for the system required the storage of all the characteristics of the target signals, which limited its potential for real life applications.

Normally, the speech signal will firstly be converted into feature vectors such as PLP, Linear predictive coding (LPC), MFCC, etc. Then, the feature vectors that carry speech information are transferred to a speech recognizer. Speech recognizers are usually based on pattern recognition algorithms, e.g. Hidden Markov

Model (HMM), Dynamic Time Warping (DTW), and Artificial Neural Network (ANN) [5, 8, 28, 51–53]. Detailed introduction of HMM will be given in Section 2.2.2. DTW is a method that allows a computer to find an optimal match between two given sequences (e.g. time series) with certain restrictions [53, 54]. The sequences are ‘warped’ non-linearly in the time dimension to determine their similarity independent of certain non-linear variations in the time dimension. This sequence alignment method is often used in time series classification [55]. ANN is a mathematical model inspired by biological neural networks. A neural network consists of an interconnected group of artificial neurons [56, 57]. In most cases, a neural network is an adaptive system changing its structure during a learning phase. Neural networks can be used for modeling complex relationships between inputs and outputs or to find patterns in data [8, 55].

Speaker enrollment is required by some systems, which means the user must provide samples of his/her speech for the training of the acoustic models before the system can be used. Other systems, which do not need the pre-training, is called speaker independent system. The input mode can be either isolated words or continuous speech. An isolated word recognition system requires that the speaker pause briefly between words, whereas a continuous speech recognition system does not. Some systems require the speaker to read from script so as to remove speech disfluencies. Language models or artificial grammars can be used to restrict the combination of words. Besides, not all the utterances that are processed by the speech recognizer are correct, so a confidence measure is usually provided as an indicator. In many practical situations, speech recognition systems may be used in noisy environments, such as in a car or in an airport, during which speech signals are corrupted by various types of background noises [3, 6, 7, 11, 58].

The performance of a speech recognition system can be measured by the recog-

dition rate, which is defined as

$$\text{RecognitionRate} = \frac{N - D - S}{N} \times 100\%,$$

where N is the total number of labels or words, D is the number of deletions, and S is the number of substitutions in the test file.

2.2 Fundamentals of Speech Recognition

Speech recognition is a multi-class sequential pattern recognition problem [7, 28]. The objective of speech recognition is to correctly convert sound utterances (or sound waves) into word sequences. To perform the recognition, two models are usually needed, i.e. acoustic model and language model [3, 59, 60]. The acoustic model allows us to evaluate how sound utterances are related to words, while the language model provides the information about the probability of all possible word sequences that could be generated from words (or vocabulary) [7, 60]. The information provided by the two models will be used to generate a statistical model that contains all possible word sequences and their corresponding prior probabilities. Then a search algorithm is used to find the most likely word sequence based on the observation. After recognition, confidence measuring module may reject the recognized words if the confidence measure of the recognized word sequence is too low [3, 7].

Figure 2.1: Speech recognition process.

To sum up the above mentioned system, there are generally six modules involved,

i.e. feature extraction, acoustic model, language model, word lexicon (vocabulary), pattern classifier, and confidence measuring [7, 61]. The relationship of these six modules are shown in Figure 2.1. A HMM model is usually designed and trained based on the the prior knowledge of acoustic model, language model, and word lexicon to perform pattern classification. In the following parts, we will describe the modules that are related to the topic of this thesis, i.e. the robustness of speech recognition systems against noise distortion. Our discussion focuses specifically on the feature extraction process and the acoustic modeling of features as they are directly related to the topic.

2.2.1 Feature Extraction

In an ASR system, the speech waveform is first pre-processed to obtain discriminative feature vectors that carry useful information about the speech. Based on these feature vectors, a speech recognizer is utilized for pattern recognition. Feature extraction is of vital importance for ASR systems since it is the first step of the recognition process and generates the input variables that the recognition algorithm works on. Without an appropriate feature extraction method, the recognition performance will be impaired, whereas a good speech feature extractor can greatly improve the performance of the ASR system. Good speech features should be able to discriminate different speech classes, such as phonemes, and remain insensitive to distractions, e.g. environmental noise, speaker variations, etc [20, 24, 60, 62, 63]. The recognition results under noisy conditions are much worse when compared to clean conditions. Therefore, robustness against noise is one key challenge in feature extraction.

Theoretical Description

There are many different feature extraction algorithms, such as Mel-Frequency Cepstral Coefficient (MFCC) [60], Perceptual Linear Prediction (PLP) coefficient [5, 60] and Linear Predictive Coefficient (LPC) [5, 18, 60]. Figure 2.2 shows the block diagram of MFCC. All the algorithms proposed in this thesis are developed based on MFCC.

Figure 2.2: Block Diagram of MFCC Feature Extraction Algorithm.

The human speech signal is slowly time varying and can be treated as near-stationary when considered under a short time frame [7, 60]. Therefore, the speech signal is usually separated into short duration blocks, called frames. Direct spectral analysis of the frames leads to high side-lobes in its frequency spectrum, which is called spectral leakage. Therefore, after partitioned into frames, the speech signal (each frame) is multiplied by a window function prior to the spectral analysis to reduce the discontinuity effect introduced by the framing process [64]. The spectral coefficients of the speech frames are estimated using the fast Fourier transform (FFT) algorithm. The spectrum of speech signal is then filtered by a group of triangular bandpass filters that simulate the characteristics of the human auditory system. These windows are called the Mel windows. The Mel filtering is to model the human auditory system that perceives sound in a nonlinear frequency binning. The mapping relationship between Hz and Mel is represented by

$$Mel(f) = 2595 \times \log_{10} \left(1 + \frac{f}{700} \right). \quad (2.1)$$

The base of each triangular filter is determined by the centre frequencies (defined in Equation (2.1)) of the neighbouring filters, and all filters are of the same height (amplitude). The bandwidth of the window is narrow at low frequencies and gradually increases for higher frequencies [7]. Figure 2.3 gives an example of the Mel-scaled triangular filter bank [65].

Figure 2.3: Mel-scaled triangular filter bank.

While the Mel filtering approximates the nonlinear characteristics of the human auditory system in frequency, the natural logarithm deals with the loudness nonlinearity. It approximates the relationship between a human's perception of loudness and the sound intensity [66]. The DCT is applied on the log Mel filterbank coefficients to generate the cepstral coefficients, and this process is modified Homomorphic processing [64].

In addition to the normal MFCC features, the energy of the speech frame is also used as part of the speech features. The log energy, called logE, is calculated directly from the time domain signal of each frame. Sometimes, it is replaced by C_0 , the 0-th component of the MFCC feature (DCT coefficients), which is the sum

of the log Mel filterbank coefficients [7, 60].

Mathematical Description

Since Cepstral Mean & Variance Normalization (CMVN), which is implemented as part of the algorithms proposed in this thesis, is well known to be able to remove convolutive noise [47], only additive noise is taken into account in further discussion. Following the standard discrete-time notation, we can obtain the linear system model for the acoustic distortion in the time and frequency domains [45, 67]. Assuming that the clean speech, $x(t)$, is corrupted by the independent additive noise, $n(t)$, the resulting noisy speech waveform, $y(t)$, is shown as follows,

$$y(k) = x(k) + n(k), \quad (2.2)$$

where k is the sample index, i.e. the k -th element in the signal sequence.

The speech signal is cut into frames and transformed into the frequency domain using Discrete Fourier Transform (DFT). Equation (2.2) becomes

$$Y(f, t) = X(f, t) + N(f, t), \quad (2.3)$$

where $Y(f, t)$, $X(f, t)$ and $N(f, t)$ refer to the spectral domain signal of noisy speech, clean speech and additive noise, respectively; f is the frequency index, i.e. the f -th element in the DFT domain; t is the frame index, i.e. the t -th frame.

Then, the power spectrum of the noisy speech is given by

$$|Y(f, t)|^2 = |X(f, t)|^2 + |N(f, t)|^2 + 2|X(f, t)||N(f, t)| \cos[\theta(f, t)]. \quad (2.4)$$

Define

$$|\tilde{N}(f, t)|^2 = |N(f, t)|^2 + 2|X(f, t)||N(f, t)| \cos[\theta(f, t)], \quad (2.5)$$

where $\theta(f, t)$ denotes the (random) angle between the two complex variables, $X(f, t)$ and $N(f, t)$. Then

$$|Y(f, t)|^2 = |X(f, t)|^2 + |\tilde{N}(f, t)|^2. \quad (2.6)$$

After applying Mel-filterbanks to the power spectra,

$$\sum_f W_f^l |Y(f, t)|^2 = \sum_f W_f^l |X(f, t)|^2 + \sum_f W_f^l |\tilde{N}(f, t)|^2, \quad (2.7)$$

where W_f^l stands for the transfer function for the l -th filter.

Changing Equation (2.7) to the log-power domain,

$$\log \left[\sum_f W_f^l |Y(f, t)|^2 \right] = \log \left[\sum_f W_f^l |X(f, t)|^2 + \sum_f W_f^l |\tilde{N}(f, t)|^2 \right], \quad (2.8)$$

where $\log(\cdot)$ is the natural logarithm.

Finally, the MFCCs can be calculated by

$$C(f_c, t) = DCT \left\{ \log \left[\sum_f W_f^l |Y(f, t)|^2 \right] \right\}, \quad (2.9)$$

where f_c is the cepstral coefficient index.

2.2.2 Hidden Markov Model

So far, many different types of recognizers have been proposed. Typical recognizers are based on DTW (Dynamic Time Warping), HMM (Hidden Markov Model), Artificial Neural Network (ANN) and SVM (Support Vector Machine) [60]. Under different conditions, they can perform very well and sometimes they can be combined to achieve better performance. Modern speech recognition systems are generally based on HMM.

Due to its rich mathematical structure and theoretical basis as well as high practicability for applications, Hidden Markov Models (HMM) are widely adopted

in ASR systems [18,68]. Hidden Markov Model is an extension of Markov Chain. In a regular Markov model, each state is visible and the state transition probabilities from one state to another are the only parameters. In a Hidden Markov Model, the state is not directly visible. Each state has a probability distribution over the possible output tokens [10,26]. Therefore, the sequence of tokens generated by a HMM provides information about the hidden sequence of states. The challenge of HMM is to determine the hidden sequence of states based on the sequence of tokens.

An isolated-word recognition process can be used as an example for explanation [59]: an N -state HMM is used for each word of a vocabulary of W . Assuming that the coding is done by using a spectral codebook with M unique spectral vectors, each observation is the index of the spectral vector closest (in some spectral sense) to the original speech signal. Thus, for each vocabulary word, a training sequence consisting of a number of repetitions of sequences of codebook indices of the word (by one or more speakers) is obtained. The first step is to optimally estimate model parameters for each word model and build individual vocabulary word models. Secondly, each of the word training sequence is segmented into states and then the properties of the spectral vectors that lead to the observation occurring in each state can be studied. Finally, after the HMM models are optimized (via a training process), recognition of an unknown word is done by scoring each word model based on the given test observation sequence (the codebook of each test word) and select the word with the highest likelihood [59,68].

The fundamental rule in statistical speech recognition systems is Bayesian decision rule which is based on the posterior probability $p(w_1^M | c_1^T)$ of a word sequence, $w_1^M = w_1, \dots, w_M$, given a sequence of acoustic observations (MFCC feature vectors), $C_1^T = \mathbf{C}(1), \mathbf{C}(2), \dots, \mathbf{C}(T)$. M is the total number of ASR output words and T is the total number of input speech frames. The word sequence $\{w_1^M\}_{opt}$ which

Figure 2.4: The Markov Generation Model (taken from HTKbook).

maximizes this posterior probability also minimizes the probability of an error in the recognized sentence [5, 59, 69].

$$\begin{aligned} \{w_1^M\}_{opt} &= \arg \max_{w_1^M} P(w_1^M | \mathbf{C}_1^T) \\ &= \arg \max_{w_1^M} \frac{P(\mathbf{C}_1^T | w_1^M) P(w_1^M)}{P(\mathbf{C}_1^T)}, \end{aligned} \quad (2.10)$$

where $P(w_1^M)$ denotes the language model probability; $P(\mathbf{C}_1^T | w_1^M)$ is the acoustic model probability; $P(\mathbf{C}_1^T)$ is the probability of the acoustic observations.

In HMM based speech recognition, it is assumed that the sequence of observed speech vectors corresponding to each word is generated by a Markov model as shown in Figure 2.4 [59]. The joint probability that $\mathbf{C}_1^T$ is generated by the model H_m moving through the state sequence $\mathbf{X}$ is calculated simply as the product of the transition probabilities and the output probabilities [51, 59].

$$p(\mathbf{C}_1^T, \mathbf{X} | H_m) = a_{12}b_2(\mathbf{o}_1)a_{22}b_2(\mathbf{o}_2)a_{23}b_3(\mathbf{o}_3) \dots \quad (2.11)$$

In practice, only the observation sequence $\mathbf{C}_1^T$ is known and the underlying state sequence $\mathbf{X}$ is hidden [59]. Given that $\mathbf{X}$ is unknown, the required likelihood is

computed by summing over all possible state sequences $\mathbf{X} = x(1), x(2), \dots, x(T_s)$, that is

$$P(c_1^T | M) = \sum_X a_{x(0)x(1)} \prod_{t=1}^{T_s} b_{x(t)}(C_t) a_{x(t)x(t+1)}, \quad (2.12)$$

where T_s is the total number of the state sequence.

In this thesis, the HMM software toolkit called HTK (Hidden Markov Model Toolkit) is used for the experiments. HTK is the Hidden Markov Model Toolkit developed by the Speech Vision and Robotics Group of Cambridge University Engineering Department (CUED) in 1989 [70]. This toolkit aims at building and manipulating Hidden Markov Model (HMM). HTK consists of a set of library modules and tools available in C source code and is primarily used for speech recognition research. The tools provide sophisticated facilities for speech analysis, HMMs training, testing and results analysis [59, 59, 71].

2.3 Frontend Noise Removal Algorithms

Noise robustness is one of the most important objectives in speech recognition. ASR systems consist of two different parts, feature extraction and pattern matching. State-of-the-art statistical speech recognizer can work very well with clean speech. Therefore, noise reduction is a straight forward approach to achieve the above mentioned objective. There are different ways to remove noise. Ephraim et al. derived the short-time spectral amplitude (STSA) estimator using Minimum Mean Square Error (MMSE) in 1984 [36], which has become a standard approach for clean speech estimation in speech processing. The advantage of a MMSE estimator is very obvious. It is mathematically optimized, which theoretically can get the optimal estimation of the clean speech. Besides, there is solid derivation making it easier to analyze. Originally, the MMSE based algorithms are intended for speech

enhancement.

For speech recognition, several MMSE based algorithms have been developed. Yu et al. in 2008 developed the MMSE estimator in the log-power domain [39]. The cepstral domain estimator is also published in 2008 [72]. Besides, different distortion models are developed to improve the performance of ASR systems [40, 72–75]. Recently, more complicated MMSE based algorithms which require the so called stereo data input have been proposed [45]. Admittedly, MMSE works well for speech enhancement and speech recognition. The main idea of MMSE is to estimate the clean speech from the noisy speech. The success of MMSE in previous implementation demonstrates that it is one of the effective means to improve the performance of ASR systems.

Figure 2.5: Different Domains in MFCC Feature Extraction Frontend.

Although there are different kinds of feature domain methods, MFCC based feature domain methods mainly involve modifications in three different domains as shown in Figure 2.5. Depending on the location of their implementation, the filters can be divided into three types, time domain filter, time-frequency domain filter and cepstral domain filter. For time domain, temporal masking is a good example, including forward masking and backward masking [2, 76]. Time-frequency domain approaches include 2D psychoacoustic filter [49], auditory model [77], Lateral Inhibition [78], Spectral Subtraction [38] and Wiener filtering [40]. Cepstral Domain

approaches include MVA [47], RASTA [79] as well as Cepstral Mean and Variance Normalization [80].

2.3.1 Spectral Filtering

MFCC is one of the most commonly used speech features. In the MFCC feature extraction algorithm, the speech signal is first cut into frames and transferred into the spectral domain using Discrete Fourier Transform (DFT) [60, 81]. Then we obtain the time-frequency representation of speech signal [6, 60, 82–85]. The DFT representation is more suitable for stationary signals [82, 83]. However, speech signal is quasi-stationary, which means that speech temporal and spectral characteristics change over time. Therefore, in speech processing, the DFT is applied to short speech segments (frames). It is assumed that during the short period (e.g. 10-30 ms) properties of speech do not change significantly. This is the so-called Short-time Fourier Transform (STFT) [60].

Frequency Domain Additive Noise Model

Since Fourier transform is a linear transform, the additive noise model is still applicable in the frequency domain. Therefore the noisy speech can be expressed in the frequency domain as follows,

$$Y(f, t) = X(f, t) + N(f, t), \quad (2.13)$$

where $Y(f, t)$, $X(f, t)$ and $N(f, t)$ are the complex Fourier transformed signals of noisy speech, $y(t)$, clean speech, $x(t)$, and noise signal, $n(t)$, respectively; f is the frequency index.

Background noise is assumed to be stationary. Thus the expectation of noise spectrum can be estimated and updated during periods when speech is absent.

Spectral Subtractive Filtering

The spectral subtractive filtering is one of the simplest algorithms proposed for noise reduction. The basic principle is described in [38]. An estimate of the clean speech spectrum is obtained by subtracting an estimate of the noise spectrum from the noisy speech spectrum. This process can be expressed as follows,

$$\tilde{X}(f, t) = \max[|Y(f, t)| - E(|N(f, t)|), 0] e^{i\theta_Y(f, t)}, \quad (2.14)$$

where $\tilde{X}(f, t) = |\tilde{X}(f, t)|e^{i\theta_Y(f, t)}$ is a complex number with noisy phase information, i is the imaginary unit defined as

$$i^2 = -1,$$

$\theta_Y(f, t)$ is the phase information of noisy speech which is not processed during noise reduction filtering in most cases, $E(\cdot)$ is the expectation function and $E(|N(f, t)|)$ is the average noise magnitude. Ideally, it should be $|N(f, t)|$ instead of $E(|N(f, t)|)$ but there is no way of obtaining it. The half-wave rectification process is carried out by the $\max(\cdot)$ function. It is one of the many ways (such as full-wave rectification) of ensuring the nonnegative property of $|\tilde{X}(f, t)|$.

Spectral Subtraction (SS) can also be implemented on the power spectra. The magnitude spectral subtraction algorithm can be easily extended to the power spectrum domain by multiplying Equation (2.13) with the conjugate $Y^*(f, t)$. By assuming the noise is zero mean and uncorrelated with the clean speech, the estimate of clean speech power spectrum can be obtained as follows

$$|\tilde{X}(f, t)|^2 = \max\{|Y(f, t)|^2 - E[|N(f, t)|^2], 0\}. \quad (2.15)$$

Similar to $E[|N(f, t)|]$, $E[|N(f, t)|^2]$ can be estimated and be used to replace the unknown part $|N(f, t)|^2$. Half-wave rectification is also used to filter off the negative values. Equation (2.15) thus describes the power spectral subtraction algorithm.

From the above discussion, it can be seen that spectral subtractive algorithms are simple to implement. However, this simple subtraction comes at a price of introducing a distortion in the enhanced speech known as musical noise (or musical tone). The technique proposed in [86] mitigates the musical noise distortion to some extent by using power spectral subtraction together with the a-priori signal to noise ratio. Different variations of spectral subtraction have been developed over the years. The most common variations involve the use of an over-subtraction factor that controls the amount of speech spectral distortion caused by the subtraction process. Different methods have been proposed for computing the over-subtraction factor based on different criteria that include linear [87] & nonlinear [88] functions, psychoacoustic masking thresholds [89], etc.

Minimum Mean Square Error Filtering

The Minimum Mean Square Error (MMSE) Short-Time Spectral Amplitude (STSA) estimator was first reported by Ephraim and Malah in 1984 [36]. The algorithm models speech and noise spectral components as statistically independent Gaussian random variables. By minimizing the mean square error, the problem is formulated as

$$\min \left[|Y(f, t)| - |\tilde{X}(f, t)| \right]^2. \quad (2.16)$$

The enhanced speech is obtained by the following equation

$$\begin{aligned} & \tilde{X}(f, t) \\ = & \Gamma(1.5) \frac{\sqrt{v(f, t)}}{\gamma(f, t)} M[-0.5; 1; -v(f, t)] Y(f, t) \\ = & \Gamma(1.5) \frac{\sqrt{v(f, t)}}{\gamma(f, t)} \exp \left[-\frac{v(f, t)}{2} \right] \times \\ & \left\{ [1 + v(f, t)] I_0 \left[\frac{v(f, t)}{2} \right] + v(f, t) I_1 \left[\frac{v(f, t)}{2} \right] \right\} Y(f, t), \end{aligned} \quad (2.17)$$

where $\Gamma(\cdot)$ denotes the gamma function; $M(a; c; x)$ is the confluent hypergeometric function; $I_0(\cdot)$ and $I_1(\cdot)$ denote the zero and first order modified Bessel function; $\xi(f, t)$ and $\gamma(f, t)$ are the a-priori and a-posteriori signal-to-noise ratios (SNR), respectively [36].

$$v(f, t) = \frac{\xi(f, t)}{1 + \xi(f, t)} \gamma(f, t). \quad (2.18)$$

$\xi(f, t)$ and $\gamma(f, t)$, are defined by

$$\xi(f, t) = \frac{\lambda_x(f, t)}{\lambda_n(f, t)}, \quad (2.19)$$

$$\gamma(f, t) = \frac{|Y(f, t)|^2}{\lambda_n(f, t)}, \quad (2.20)$$

where $\lambda_x(f, t) \triangleq E[|X(f, t)|^2]$, $\lambda_n(f, t) \triangleq E[|N(f, t)|^2]$ are the variances of the spectral component of the speech and the noise, respectively.

Function $M(; ;)$ in Equation (2.17) is the confluent hypergeometric function which is defined in [90]. It can be computed efficiently using the following series summation:

$$M(a; b; c) = \sum_{m=0}^{\infty} \frac{(a)_m c^m}{(b)_m m!} = 1 + \frac{ac}{b1!} + \frac{a(a+1)c^2}{b(b+1)2!} + \frac{a(a+1)(a+2)c^3}{b(b+1)(b+2)3!} + \dots \quad (2.21)$$

Although the principle based on the squared error of the magnitude spectra is mathematically tractable, it may not be subjectively meaningful to the human auditory system [36]. The relationship between a human's perception of loudness and the sound intensity is nonlinear [64, 66]. Therefore, the natural logarithm is usually utilized to deal with the loudness nonlinearity. It has been suggested that a principle based on the squared error of the log-magnitude spectra may be more suitable for speech processing [37, 91, 92]. An estimator that minimizes the mean square error of the log-magnitude spectra is derived in [37]. The problem is formulated as

$$\min \left[\log |Y(f, t)| - \log |\tilde{X}(f, t)| \right]^2. \quad (2.22)$$

This optimal log-MMSE estimator is given by the following equation:

$$|\tilde{X}(f, t)| = \frac{\xi(f, t)}{1 + \xi(f, t)} \exp\left(\frac{1}{2} \int_{v(f, t)}^{\infty} \frac{e^{-t}}{t} dt\right) |Y(f, t)|, \quad (2.23)$$

where $v(f, t)$ is defined in Equation (2.20) and $\xi(f, t)$ is defined in Equation (2.19).

MMSE based speech enhancement algorithms are known to be good at noise reduction without leaving obvious annoying musical tones.

2.3.2 Subspace Filtering

Subspace filtering is a different class of algorithms that are based on linear algebra theory. More specifically, these algorithms are based on the principle that the clean signal might be confined to a subspace of the noisy Euclidean space [93, 94]. The decomposition of the vector space of the noisy signal into a signal subspace and a noise subspace can be performed by the Singular Value Decomposition (SVD), eigenvector-eigenvalue factorization or Karhunen-Loeve transform (KLT) to the noisy signal [93].

A KLT based algorithm is the process whereby KLT components are filtered by a gain function and then undergo the inverse KLT to restore the enhanced speech. Let R_x denote the covariance matrix of clean speech, $x(t)$, with K samples, the eigen-decomposition of R_x is given by:

$$R_x = U \Lambda_x U^T, \quad (2.24)$$

where $\Lambda_s = \text{diag}[\lambda_x(1), \lambda_x(2), \dots, \lambda_x(K)]$ is the eigenvalue matrix; U is the eigenvector matrix of R_x ; U^T is the transpose of U .

As analyzed in [93], because for speech signals the number of basis vectors is smaller than the dimension of the vector, some of the eigenvalues of the covariance matrix must be zero, which means the covariance matrix of clean speech is not full

rank. On the other hand, all eigenvalues of white noise are positive. In other words, the covariance matrix of noise is usually full rank. Hence, the vector space of a noisy speech is composed of a signal-plus-noise subspace, which should be processed by speech enhancement algorithms and a noise dominated subspace which can be simply discarded [6,93,95]. According to the spectral domain constrained estimator proposed in [93], a linear filter of the subspace speech enhancement method H_{sub} can be obtained by

$$H_{\text{sub}} = UGU^T = U \begin{bmatrix} G_\mu & 0 \\ 0 & 0 \end{bmatrix} U^T. \quad (2.25)$$

G_μ is a full rank $M \times M$ matrix where M is the assumed number of non-zero eigenvalues of the autocorrelation matrix R_x . A possible solution of the spectral domain constrained estimator is

$$G = \Lambda_x (\Lambda_x + \sigma_n^2 \Lambda_\mu)^{-1}, \quad (2.26)$$

where σ_n^2 is the noise variance; $\Lambda_\mu = \text{diag}[\mu(1), \mu(2), \dots, \mu(K)]$ is a diagonal matrix of Lagrange multipliers when deriving the estimator, H_{sub} , based on the criterion of minimizing the speech signal distortion with a permissible residual noise level [93]. The k th diagonal element of G is given by:

$$g(k) = \begin{cases} \frac{\lambda_x(k)}{\lambda_x(k) + \mu(k)\sigma_n^2}, & k = 1, 2, \dots, M \\ 0, & k = M + 1, \dots, K \end{cases}. \quad (2.27)$$

The algorithm given above is proposed by Ephraim [93], and is based on the white noise model. In 2000, Mittal et al. proposed the so-called signal/noise KLT-based approach for the colored noise condition in [96]. An adaptive algorithm [95] can also be used to solve this problem.

2.3.3 Stereo Data Based Noise Removal

The stereo data based algorithms have very promising ability to recover clean speech. Stereo data contains both the clean speech and the corresponding noisy speech. One of the most important features of the stereo data based algorithms is that the statistical recognizer (HMMs) is trained merely on the original clean speech [16, 45]. Normally, for most of the noise robust algorithms, the training materials have to be processed by the same noise removal algorithm to make up for the distortion caused by themselves. However, this kind of algorithms don't need the compensation procedure, which means the processed speech is much closer to the clean speech. Stereo-based Piecewise Linear Compensation for Environments (SPLICE) is a good example. It produces an estimate of cepstrum of undistorted speech given the observed cepstrum of distorted speech [16].

Given the general model of distortion from a clean cepstral vector, C_x , into a noisy one, C_y , the probabilistic formulation of the basic (frame independent) version of the SPLICE algorithm can be described as follows. The first assumption is that the noisy speech cepstral vector follows the distribution of a mixture of Gaussians [16].

$$p(C_y) = \sum_s p(C_y|s) p(s), \quad (2.28)$$

where $p(C_y|s)$ is assumed to be

$$p(C_y|s) = N(C_y; \mu_s, \Sigma_s), \quad (2.29)$$

where s is the state index of GMM.

The second assumption made by SPLICE is that the conditional probability density function (PDF) for the clean vector, C_x , given the noisy speech vector, C_y ,

and the region index, s , is Gaussian whose mean vector is a linear transformation of the noisy speech vector, C_y [16]. The conditional PDF is assumed to have the form,

$$p(x|y, s) = N(x; y + r_s, \Gamma_s). \quad (2.30)$$

The MMSE is the following conditional expectation of clean speech vector given the observed noisy speech [16]

$$\hat{x}_{MMSE} = E_x(x|y) = \sum_s p(s|y) E_x(x|y, s). \quad (2.31)$$

It is clear that

$$E_x(x|y) = y + r_s. \quad (2.32)$$

Then

$$\hat{x}_{MMSE} = y + \sum_s p(s|y) r_s. \quad (2.33)$$

Since the noisy speech PDF, $p(y)$, is assumed to be a mixture of Gaussians, the standard EM algorithm can be used to train μ_s and Γ_s on noisy speech [16]. Initial values of the parameters are determined by a VQ clustering algorithm. If stereo data is available, the parameters, r_s , of the conditional PDF, $p(x|y, s)$, can be trained using the maximum likelihood criterion

$$r_s = \frac{\sum_n p(s|y_n) (x_n - y_n)}{\sum_n p(s|y_n)}, \quad (2.34)$$

$$p(s|y_n) = \frac{p(y_n|s) p(s)}{\sum_s p(y_n|s) p(s)}. \quad (2.35)$$

This training procedure requires a set of stereo (two channels) data. One channel contains the clean utterance and the other contains the same utterance with distortion.

2.3.4 Cepstral Filtering

Clean speech estimation is one of the most popular front-end noise removal algorithms. All the above mentioned algorithms belong to this category. However, there is also another kind of approach which uses simple smoothing methods and manages to obtain very promising results. Mean subtraction, variance normalization & ARMA filtering (MVA) [47] is a good example. It simply normalizes or smoothes the temporal structure of the speech feature.

The algorithm proposes to first perform Mean subtraction (MS) and Variance normalization (VN), shown in Equation (2.36).

$$\hat{C}(f_c, t) = \frac{C(f_c, t) - \mu_c}{\sigma_c^2}, \quad (2.36)$$

where $\hat{C}(f_c, t)$ is the mean-subtracted and variance-normalized feature, μ_c is a mean vector estimated from the cepstral data and σ_c^2 is an estimate of the cepstral data variance.

Then, an auto-regression and moving-average (ARMA) filter is adopted to further process the cepstral feature.

$$\tilde{C}(f_c, t) = \frac{\tilde{C}(f_c, t - M) + \dots + \tilde{C}(f_c, t - 1) + \hat{C}(f_c, t) + \dots + \hat{C}(f_c, M)}{2M + 1}, \quad (2.37)$$

where $\tilde{C}(f_c, t)$ is the MVA post-processed feature and M is the order of the ARMA filter. The special case of $M = 0$ degenerates to no filtering.

Chapter 3

Psychoacoustic Modeling

3.1 Introduction

State-of-the-art automatic speech recognition (ASR) systems can work very well with clean speech, yielding a recognition rate of about 99% for small vocabulary tasks. However, when it comes to noisy speech, there is a dramatic drop in system performance. The human auditory system is extremely adept at handling noise [97–101]. Therefore, it is natural to study the auditory model and apply it to an automatic speech recognition (ASR) system. In this chapter, a feature extraction front-end with noise-robust characteristics based on psychoacoustic model is proposed to improve the performance of automatic speech recognition systems.

For human, hearing is not simply a mechanical phenomenon of wave indexing [18, 102, 103]. It is more like a sensory and perceptual process. Before we recognize the speech, the signal has already been processed by the human auditory system, which means the signal that we ‘hear’ is not the physical signal produced by the sound source. The basic idea of the proposed algorithm is to recognize speech based

on this ‘new’ version of speech. The science of studying how human perceives sounds including relationships between sound pressure level and loudness, human response to different frequencies and a number of masking effects is called psychoacoustics [18, 102]. Previous research shows that successful application of human auditory models can greatly improve the performance of ASR systems. The Mel-Frequency Cepstral Coefficients (MFCC) method can serve as a good example. In MFCC, an auditory-based warping of the frequency axis, called critical-band, is implemented which is so successful that MFCC has become a standard speech feature for ASR systems [104, 105]. The algorithm introduced in this chapter aims to integrate masking effects into the MFCC auditory model and study the resulting effects when applied to automatic speech recognition.

Masking effect is the phenomenon that describes how a clearly audible sound (maskee) is influenced by another sound (masker). To measure the effect of masking quantitatively, a masking threshold is usually determined. The masking threshold is the sound pressure level of a test sound, to be just audible in the presence of a masker. Masking effects may be classified as simultaneous or temporal according to the occurrence of the signals [102]. Masking effect between any two signals which occur at the same time is called simultaneous masking (a.k.a. frequency masking). In temporal asking, signals can be masked by the preceding sound, called forward masking, or by the sound after it, called backward masking. These masking effects are caused by the principal mechanism of neuronal signal processing both in time and frequency domains [76, 78, 106].

3.2 Temporal Masking

A weak sound emitted soon after the end of a louder sound can be masked by the louder sound. In fact, even a weak sound occurring just before a louder sound can also be masked by the louder sound. These two effects are called forward and backward temporal masking respectively. Forward masking (FM) reveals that over short durations, the usable dynamic range of the human auditory system depends on the spectral characteristics of the previous stimuli. Forward masking can be viewed as a consequence of auditory adaptation [107]. The parameters for forward masking are frequency dependent. Backward masking has the same characteristic but the masker comes after the signal rather than before it. Figure 3.1 shows the characteristic curve of temporal masking.

Figure 3.1: Sketch of psychoacoustic data (temporal masking).

Much work has been conducted to study temporal masking. In 1982, Jesteadt, Bacon and Lehmann presented a series of psychoacoustic experiment results for pure tone forward masking [2]. However, the longest probe delay involved was 40 ms which is not enough for complete adaptation. Some more complete experimental results and mathematical models have been proposed later. In 1997, Stroe and

Alwan proposed a more complete psychoacoustic model [107] which models temporal masking by psychoacoustic adaptation.

Models of adaptation have been successfully applied in ASR [41, 107–110]. Adaptation and the popular RASTA techniques have obvious similarities to one another [79]. However, forward masking has a recovery step after adaptation as reported by Oxenham et al. [1]. Stroppe and Alwan [107] have demonstrated forward masking and its application to ASR. Tchorz and Kollmeier [108] tested various adaptation parameters in ASR systems. Recently, an adaptation model has been developed and evaluated based on AURORA2 and AURORA3 [111].

3.2.1 Type I Forward Masking Modeling

This type of forward masking model is based on three equations. The first, Equation (3.1), is for characterizing recovery, i.e. upward adaptation. It describes the model's prediction of the forward-masking threshold with long maskers as a function of masker level and probe delay. The second equation, Equation (3.2), which takes into account masker duration, is to characterize an attack and the last is to calculate the probe threshold difference, i.e. downward adaptation. It describes the case where the model reaches the static threshold before the onset of the masker.

Equation (3.1) is a first order difference equation which leads to an exponential decay of the logarithmic distance to the target [107, 112, 113].

$$P = M(1 - m)a^u, \quad (3.1)$$

where P is probe level (describes the amount of masking), M is masker level, u is discrete-time probe delay and m is input/output (I/O) slope (see [107]). As the duration between masker and probe, u , increases, the amount of masking effect imposed on the incoming probe, P , decreases. This equation is used to determine

Table 3.1: Forward Masking Model Parameters.

Freq. (Hz)	Slope m	a	b
250	0.19	0.864	0.474
500	0.20	0.854	0.510
1000	0.26	0.816	0.543
2000	0.29	0.851	0.525
4000	0.34	0.858	0.507

the model parameters m as well as a .

In the case of downward adaptation, Equation (3.2) assumes incomplete adaptation and describes the amount of masking as a function of attack parameter, b , and discrete time masker duration, nd , as well as probe delay, nu ,

$$P = M(1 - m)(1 - b^{nd})a^{nu}. \quad (3.2)$$

The difference between Equations (3.1) and (3.2) is due to the assumption about whether complete adaptation has already occurred. The attack parameter, b , is determined through the following probe threshold difference equation,

$$\Delta P = M(1 - m)b^{nd}a^{nu}. \quad (3.3)$$

The model parameters (m , a , and b) discussed above varies with the frequency of the input signal. Five sets of data corresponding to the frequencies at 250 Hz, 500 Hz, 1000 Hz, 2000 Hz and 4000 Hz are given in Table 3.1 [107]. The importance of these constants is that their ratio approximates the psychoacoustic data from physiological data presented by Goldhor in 1985 [114,115].

The frequency limits for the filter bank range from 0 Hz up till 8 kHz for the evaluation experiments. For the Mel scaled filter bank, the centre frequencies are important as they are used to calculate the model parameters for determining the dynamic model. The centre frequencies for 24 Mel filter banks from 0 Hz to 8 kHz

Table 3.2: Centre Frequencies for Mel Scaled Filter Banks.

Filter No.	1	2	3	4	5	6	7	8
f_c (Hz)	46.88	152.34	230.47	320.31	414.06	519.53	632.81	761.72
Filter No.	9	10	11	12	13	14	15	16
f_c (Hz)	46.88	152.34	230.47	320.31	414.06	519.53	632.81	761.72
Filter No.	17	18	19	20	21	22	23	24
f_c (Hz)	2589.84	2902.34	3242.19	3617.19	>4000	>4000	>4000	>4000

are tabulated in Table 3.2. It is easy to tell from Table 3.2 that only 20 filter banks are useful in the evaluation experiment due to the Nyquist theory.

The model parameters (a , b , and m) for each filter bank are calculated based on the centre frequency of each filter bank. Every filter bank will share a set of model parameters. Since only five sets of model parameters are available, linear interpolation is required to derive the model parameters for all the filter banks. The weights are determined using the distances between the centre frequency, the two neighboring frequencies and the corresponding parameters at these two frequencies, as follows

$$V = \frac{w_1 \cdot v_1 + w_2 \cdot v_2}{w_1 + w_2}, \quad (3.4)$$

$$w_1 = \frac{f_2 - f_c}{f_2 - f_1}, \quad (3.5)$$

$$w_2 = \frac{f_c - f_1}{f_2 - f_1}, \quad (3.6)$$

where f_1 and f_2 take on either one of the 5 frequencies (250 Hz, 500 Hz, 1000 Hz, 2000 Hz and 4000 Hz) and $f_2 > f_1$. Thus, values of f_1 and f_2 correspond to one of five frequencies that is just below the centre frequency and one that is just above the centre frequency. The values, v_1 and v_2 , are the model parameters at f_1 and f_2 .

3.2.2 Type II Forward Masking Modeling

Other than the first type of forward masking modeling, a first order infinite impulse response (IIR) high-pass filter can be used as the Type II forward masking model [116]. The transfer function of the high-pass filter is

$$H(z) = \frac{2f_s\tau - 2f_s\tau z^{-1}}{1 + 2f_s\tau + (1 - 2f_s\tau)z^{-1}}. \quad (3.7)$$

The relationship between input and output can be written as

$$y_o(t) \cdot (1 + 2f_s\tau) = 2f_s\tau \cdot y_{in}(t) - 2f_s\tau \cdot y_{in}(t-1) - (1 - 2f_s\tau) \cdot y_o(t-1), \quad (3.8)$$

$$y_o(t) = \frac{2f_s\tau}{1 + 2f_s\tau} y_{in}(t) - \frac{2f_s\tau}{1 + 2f_s\tau} y_{in}(t-1) - \frac{1 - 2f_s\tau}{1 + 2f_s\tau} y_o(t-1), \quad (3.9)$$

where y_{in} and y_o are the input and output speech signal (noisy speech) respectively, $f_s = 128$ Hz, the MFCC frame rate in the evaluation experiments and time constant $\tau = 240$ ms.

The filter's memory is initialized to zero and the first frame of logarithmic Mel spectra is subtracted from all the frames when applying this filter to avoid an initial transient.

3.3 Simultaneous Masking

Psychoacoustics is the scientific study of sound perception. Simultaneous masking studies how the amount of masking changes with frequency. The amount of masking refers to the difference in power level between the actual speech and what is perceived by humans. Usually it is defined as a subtractive energy level. Simultaneous masking has been widely used in many areas such as audio compression and speech enhancement [115, 117–119]. However, its implementation in speech recognition is

relatively new. Lateral Inhibition (LI) is one effective approach for implementing simultaneous masking [49, 77, 78, 106, 118].

Figure 3.2: Sketch of psychoacoustic data (frequency masking).

In neurobiology, lateral inhibition is used to measure the capacity of an excited neuron to reduce the activity of its neighbors. It is a common phenomenon in sensory reception of biological systems. In speech processing, it stands for how two speech signals with different frequencies but the same commencing time affect each other. LI helps to sharpen spectral changes. Figure 3.2 shows the sketch of the psychoacoustic data given in Houtgast's paper [117, 120]. The curve describes how the amount of LI masking changes with frequency.

Assume that a speech signal, $x(k)$, is corrupted by noise, $n(k)$, resulting in a noisy speech, $y(k)$. The relationship is given by

$$y(k) = x(k) + n(k), \quad (3.10)$$

where k is the sample index as defined in Chapter 2. The speech signal is cut into frames and transformed into frequency domain using DFT. Then Equation (3.10)

becomes

$$Y(f, t) = X(f, t) + N(f, t), \quad (3.11)$$

where f is the frequency index; t is the frame index; $Y(f, t)$, $X(f, t)$, $N(f, t)$ refer to the time-frequency domain signal of noisy speech, clean speech and additive noise.

By assuming the additivity on the powers of the components in the frequency domain [18, 121], the power spectrum of the noisy speech becomes

$$|Y(f, t)|^2 = |X(f, t)|^2 + |\tilde{N}(f, t)|^2. \quad (3.12)$$

Let $M_{LI}(f)$ represent the lateral inhibition masker. The lateral inhibition masker is modeled to satisfy the following constraint [102, 117],

$$\int_{-\infty}^{\infty} M_{LI}(f) df = 0. \quad (3.13)$$

The LI masker is very effective in removing stationary noise. Assuming $|\tilde{N}(f, t)|^2$ in Equation (3.12) is stationary, after applying the given lateral inhibition masker in Equation (3.13) to the noisy speech, the processed speech, $|\tilde{Y}(f, t)|^2$, can be calculated by

$$\begin{aligned} & |\tilde{Y}(f, t)|^2 \\ &= \int_{-\infty}^{\infty} |Y(f, t)|^2 M_{LI}(f) df \\ &= \int_{-\infty}^{\infty} |X(f, t)|^2 M_{LI}(f) df + \int_{-\infty}^{\infty} |\tilde{N}(f, t)|^2 M_{LI}(f) df \\ &= |\tilde{X}(f, t)|^2 + \int_{-\infty}^{\infty} |\tilde{N}(f, t)|^2 M_{LI}(f) df \\ &= |\tilde{X}(f, t)|^2. \end{aligned} \quad (3.14)$$

Theoretically, the power spectrum of processed speech with white noise equals to that of the input clean speech. The algorithm not only sharpens the spectrum of the input signal but also removes noise.

3.4 Temporal Spectral Average (TSA)

Lateral inhibition is very sensitive to spectral changes and it will enhance the peaks in both speech signals and noise signals. Noise signals which do not have fairly stationary segments in the power spectrum such as subway noise, will be enhanced together with speech signals, thereby greatly degrading the recognition results. Hence, the temporal temporal average approach is proposed to alleviate this effect,

$$\left| \tilde{Y}(f, t) \right|^2 = \frac{1}{2N_a + 1} \sum_{\Delta t = -N_a}^{N_a} Z_{\Delta t} |Y(f, t + \Delta t)|^2. \quad (3.15)$$

Temporal averaging is a very commonly used technique in speech processing [38, 47, 122, 123]. For example, in Spectral Subtraction (SS) [38], Boll proposed to use local averaging of spectral magnitudes to reduce the spectral error, since the spectral error equals the difference between the noise spectrum and its mean. In our present implementation, the parameters are set as reported in [117], $N_a = 2$ and the smoothing parameter $Z_{\Delta t}$ is given below,

$$\begin{bmatrix} Z_{-2} & Z_{-1} & Z_0 & Z_1 & Z_2 \end{bmatrix} = \begin{bmatrix} 0.4 & 1.3 & 1.6 & 1.3 & 0.4 \end{bmatrix}.$$

3.5 Results and Discussion

3.5.1 Implementation Details

The proposed front-end feature extractor is modified from the MFCC script provided by Voicebox [124] by integrating Lateral Inhibition (LI), Temporal Masking (TM), and Temporal Spectral Average (TSA). The parameters given in Park's paper [116] are used for temporal masking. The 1D Mexican hat introduced given in Equation (3.16) is used for Lateral Inhibition (LI) [49]. The proposed algorithm

involves sequential implementation of Lateral Inhibition (LI), Temporal Masking (TM), Temporal Spectral Average (TSA), and Cepstral Mean & variance Normalization (CMVN). Since Forward Masking (FM) is dominant in temporal masking (stronger masking and longer effective range than backward masking), the algorithm is called LTFC short for LI, TSA, FM, and CMVN. The block diagram of the proposed algorithm is given in Figure 3.3.

$$[-0.07 \quad -0.27 \quad -0.16 \quad 1 \quad -0.16 \quad -0.27 \quad -0.07] \quad (3.16)$$

Figure 3.3: Block Diagram of LTFC.

The same recognizer is used for both the proposed front-end feature extraction algorithm and the baseline system for a meaningful comparison. Each digit is modeled by a simple left-to-right 18-state (including two non-emitting states) HMM model, with 3 Gaussian mixtures per state. Two pause models are defined. One is "sil", which has 3 HMM states and models the pauses before and after each utterance. The other one is "sp", which is a single state model (tied with the middle state of "sil") and models the pauses among words.

3.5.2 Experimental Results

Evaluation test is performed using the AURORA2 database. Recognition results are averaged over the noisy test sets with SNRs from 0 dB to 20 dB, which is denoted

Table 3.3: Recognition Results of LTFC for Clean Training Condition (%).

	Noise Type	Clean	20	15	10	5	0	-5	Avg 0-20
Set A	Subway	99.08	97.85	95.92	92.66	84.43	66.23	35.43	87.42
	Babble	98.97	98.13	96.92	94.11	84.22	58.89	28.96	86.45
	Car	99.19	98.33	96.75	93.38	83.63	59.23	27.05	86.26
	Exhibition	99.20	97.44	94.63	89.48	78.19	56.77	32.98	83.30
Set B	Restaurant	99.08	98.34	97.48	94.50	86.67	64.05	33.71	88.21
	Street	98.97	97.43	96.10	91.72	81.80	60.61	31.44	85.53
	Airport	99.19	98.39	97.32	95.35	86.46	62.66	29.85	88.04
	Train	99.20	97.93	96.85	93.27	84.70	61.71	29.22	86.89
Set C	Restaurant	98.96	96.96	94.47	89.93	78.42	55.82	24.56	83.12
	Street	98.88	96.98	94.89	89.42	77.33	55.53	26.78	82.83
Avg		99.07	97.78	96.13	92.38	82.59	60.15	30.00	85.81

as Avg 0-20. Tables 3.3 and 3.4 present the experimental results for the proposed LTFC (LI+TSA+FM+CMVN) algorithm. The performance metric used is word recognition rate in percentage. Results are captured for both the clean and multi training conditions.

The first set of comparison is made against MFCC, FM, LI, TSA, FM+LI+TSA, and CMVN. The second set of comparison is made against three state-of-the-art front-end noise removal algorithms, RASTA [79], MVA [47], and the temporal frequency warped 2D psychoacoustic filter [102]. Tables 3.3 and 3.4 give the detailed experimental results for the clean training condition and multi training condition, respectively. Since the recognition results for the clean subset are all about 99%, it is meaningless to discuss improvement at such level. Only the experimental results for noisy test sets are used for further discussion.

Table 3.4: Recognition Results of LTFC for Multi Training Condition(%).

	Noise Type	Clean	20	15	10	5	0	-5	Avg 0-20
Set A	Subway	98.25	98.53	97.67	96.35	93.15	82.19	55.54	93.58
	Babble	98.19	98.61	98.19	97.22	92.62	75.21	43.56	92.37
	Car	98.15	98.51	98.00	96.51	92.54	80.17	48.46	93.15
	Exhibition	98.46	98.43	97.38	94.82	88.77	73.90	51.00	90.66
Set B	Restaurant	98.25	98.68	98.40	97.39	92.97	77.31	48.17	92.95
	Street	98.19	98.1	97.76	96.10	91.44	78.14	50.06	92.31
	Airport	98.15	98.66	98.15	97.20	93.2	80.52	48.76	93.55
	Train	98.46	98.98	98.12	96.51	92.66	79.08	51.25	93.07
Set C	Restaurant	98.28	98.25	97.36	95.79	91.31	76.94	47.96	91.93
	Street	98.28	98.28	97.82	95.53	90.15	75.79	46.25	91.51
Avg		98.27	98.50	97.89	96.34	91.88	77.93	49.10	92.51

Comparison of Different Parts of LTFC

The first set of comparison is made against different parts of the proposed LTFC algorithm. From the experimental results in Tables 3.5 and 3.6, it can be seen that FM, LI, TI, and CMVN all help to improve the performance of the speech recognition system. All of them manage to obtain better results than the baseline MFCC(39) system. The relative improvements are given in Table 3.7. The relative improvements are calculated in terms of word recognition rate. Figure 3.4 gives a better view of the experimental results. It can be observed that for all noisy conditions the proposed LTFC algorithm performs better than all the above mentioned algorithms.

Compared with the baseline MFCC system, the proposed algorithm achieves very positive results. The relative improvements for the clean training condition are 0.42% at SNR of 20 dB, 2.80% at SNR of 15 dB, 13.82% at SNR of 10 dB, 47.43% at SNR of 5 dB, 111.87% at SNR of 0 dB, and 130.06% at SNR of -5 dB. For the multi training condition, the relative improvements are 0.34%, 0.31%, 0.86%,

Table 3.5: Recognition results for comparison targets under clean training condition (%).

SNR/dB	Clean	20	15	10	5	0	-5	Avg 0-20
MFCC(39)	99.36	97.37	93.51	81.16	56.02	28.39	13.04	71.29
FM	99.03	97.02	93.91	85.89	68.24	41.65	21.30	77.34
LI	99.42	97.19	94.23	83.29	60.92	34.21	17.07	73.97
TSA	99.38	97.48	94.13	83.86	61.41	33.96	17.58	74.17
FM+LI+TSA	99.06	96.63	93.59	86.26	69.97	40.94	16.47	77.48
CMVN	99.32	96.97	94.32	87.59	71.20	38.84	13.90	77.78
RASTA	99.08	96.45	93.51	86.41	70.57	42.78	19.93	77.94
MVA	99.21	97.33	95.28	90.55	80.01	57.59	27.38	84.15

4.87%, 29.09% and 83.00% for SNR of 20 dB, 15 dB, 10 dB, 5 dB, 0 dB, and -5 dB, respectively.

Figure 3.4: Experimental results.

In the clean training condition, the proposed algorithm manages to obtain noticeable improvements. At SNR of 20 dB and 15 dB, all the above mentioned algorithms achieve about 98% recognition rate. The relative improvements are about 1% at SNR of 20 dB over FM, LI, TSA, FM+LI+TSA, and CMVN. At SNR of 15 dB the relative improvements are about 2% compared with the above mentioned compari-

Table 3.6: Recognition results for comparison targets under multi training condition (%).

SNR/dB	Clean	20	15	10	5	0	-5	Avg 0-20
MFCC(39)	99.11	98.18	97.60	95.52	87.61	60.37	26.83	87.85
FM	98.74	98.16	97.47	95.25	87.19	59.32	25.46	87.48
LI	99.13	98.19	97.62	95.53	88.06	61.93	26.59	88.26
TSA	99.24	98.38	97.82	95.50	88.03	61.20	26.93	88.19
FM+LI+TSA	98.59	98.11	97.43	95.06	87.90	64.16	28.56	88.53
CMVN	98.94	98.51	97.89	96.27	91.06	74.81	42.63	91.71
RASTA	98.60	98.41	97.57	95.86	90.23	74.31	44.75	91.27
MVA	99.04	98.49	97.87	96.15	91.13	77.49	49.04	92.22

son targets. At SNR of 10 dB, the relative improvements become larger, 2.36% over FM, 10.91% over LI, 10.16% over TSA, 2.71% over FM+LI+TI, and 5.47% over CMVN. As SNR drops further, the relative improvements become larger. At SNR of 5 dB, the relative improvements range from 16% (CMVN) to 35.57% (LI). At SNR of 0 dB and -5 dB, the relative improvements are over 50%. For CMVN, the improvement even reaches 115.83%.

For the multi training condition, the recognition rates for SNR of 20 dB, 15 dB, 10 dB, and 5 dB are all above 90%. Therefore, the relative improvements appear to be small. It has to be noted that CMVN performs very well in multi training condition. Except for SNR of 0 dB and -5 dB, the relative improvements are less than 1%. However, the proposed algorithm still performs better, especially at low SNR levels. The relative improvements are 4.13% at SNR of 0 dB and 15.15% at SNR -5 dB.

For FM, LI, TI, and FM+LI+TI, the relative improvements for SNR of 20 dB and 15 dB are all less than 1%. For SNR of 10 dB, the relative improvements are around 1%. Some are larger than 1%, such as FM at 1.14% and FM+LI+TI at 1.35%. At

Table 3.7: Relative Improvements (%).

	SNR/dB	20	15	10	5	0	-5	Avg 0-20
Clean Training	MFCC(39)	0.42	2.80	13.82	47.43	111.87	130.06	20.37
	FM	0.78	2.36	7.56	21.03	44.42	40.85	10.95
	LI	0.61	2.02	10.91	35.57	75.83	75.75	16.01
	TSA	0.31	2.12	10.16	34.49	77.12	70.65	15.69
	FM+LI+TSA	1.19	2.71	7.09	18.04	46.92	82.15	10.75
	CMVN	0.84	1.92	5.47	16.00	54.87	115.83	10.32
Multi Training	MFCC(39)	0.34	0.31	0.86	4.87	29.09	83.00	5.30
	FM	0.36	0.44	1.14	5.38	31.37	92.85	5.75
	LI	0.33	0.29	0.85	4.34	25.84	84.66	4.82
	TSA	0.13	0.08	0.88	4.37	27.34	82.32	4.90
	FM+LI+TSA	0.41	0.48	1.35	4.53	21.46	71.92	4.50
	CMVN	0	0.01	0.06	0.88	4.13	15.15	0.84

SNR of 5 dB, the relative improvements range from 4.34% to 5.38%. At SNR of 0 dB and -5 dB, very positive results are obtained. The relative improvements are all over 20% at 0 dB and over 70% at SNR -5 dB.

Comparison with Other Algorithms

The second set of comparison is made against three state-of-the-art front-end noise removal algorithms. Experimental results are given in Tables 3.5 and 3.6. The relative improvements are given in Table 3.7. It can be easily seen that the proposed algorithm successfully improves the performance of the speech recognition system. As SNR decreases, the improvement becomes larger and larger.

For the clean training condition, the advantage of the proposed LTFC algorithm can be easily observed. At SNR of 20 dB the recognition rate is 97.78%. The relative improvements over RASTA, TFW 2D, and MVA are 0.10%, 0.02%, and 0.46%,

respectively. For SNR of 15 dB, the relative improvements are 2.80%, 0.20%, and 0.89%. As SNR drops further to 10 dB, the relative improvements become 6.91%, 0.84%, and 2.02% for RASTA, TFW 2D, and MVA, respectively. For SNR of 5 dB and 0 dB, the relative improvements are 17.03% and 40.60% for RASTA, 2.70% and 6.91% for TFW 2D, 3.22% and 4.45% for MVA.

Table 3.8: Relative Improvements (%).

	SNR/dB	20	15	10	5	0	-5	Avg 0-20
Clean Training	RASTA	1.38	2.80	6.91	17.03	40.60	50.53	10.10
	TFW 2D	0.02	0.20	0.84	2.70	6.91	23.10	1.67
	MVA	0.46	0.89	2.02	3.22	4.45	9.57	1.97
Multi Training	RASTA	0.10	0.34	0.50	1.83	4.87	9.72	1.36
	TFW 2D	0.16	0.10	0.26	0.87	2.91	11.95	0.76
	MVA	0.02	0.03	0.20	0.82	0.57	0.12	0.31

For the multi training condition, as discussed in the previous sections at SNR of 20 dB to 5 dB, the recognition rates are all over 90%. Therefore, the relative improvements are small at high SNR levels. However, there are still noticeable improvements. At SNR of 20 dB the relative improvements are 0.10% over RASTA, 0.16% over TFW 2D, and 0.02% over MVA. At SNR of 15 dB, the relative improvements are 0.34% over RASTA, 0.10% over TFW 2D, and 0.03% over MVA. For SNR of 10 dB, the relative improvements are 0.50% over RASTA, 0.26% over TFW 2D, and 0.20% over MVA. For SNR of 5 dB, the relative improvements are 1.83% over RASTA, 0.87% over TFW 2D, and 0.82% over MVA. When it comes to SNR of 0 dB, the relative improvements become larger. The relative improvements are 4.87% over RASTA, 2.91% over TFW 2D, and 0.57% over MVA. At SNR of -5 dB, the largest improvements are observed, 9.72% over RASTA, 11.95% over TFW 2D, and 0.31% over MVA.

Figure 3.5: Clean training condition results for babble noise.

Bigger improvements can be obtained for certain noise types as shown in Figure 3.5, which shows the word recognition rate under different SNR levels for babble noise. It is clear that the recognition rate is improved by 20% to 30% within the middle range SNR, which is a relatively good improvement for ASR systems under noisy environment, hence vindicating that our algorithm has made the traditional recognizer more robust to the noisy condition.

To illustrate the negligible computational load of the proposed method, let us take lateral inhibition masker, Equation (3.16), as an example. Regardless of programming language, masking is implemented as a convolution process. For the proposed lateral inhibition masking, the convolution can be implemented using only 6 multiplications per point. Since there are 24 mel-filter coefficients, only 44 multiplications are required per 30-ms frame after taking endpoint conditions into account. Similarly, there are 104 multiplications for TI and 64 multiplications for FM per 30-ms frame.

3.6 Conclusion

In this chapter, a MFCC feature extraction algorithm integrated with lateral inhibition, temporal integration, forward masking and cepstral mean & variance normalization is proposed and applied to an automatic speech recognition system. The proposed algorithm is evaluated with a series of classification experiments based on HMM using the AURORA2 database. The obtained results verify that the proposed algorithm effectively improves the recognition rate under noisy environments.

Chapter 4

2D Psychoacoustic Filtering

4.1 Introduction

As described in Chapter 3, hearing is a sensory and perceptual process, which is different from simple wave indexing. One of the key features of the human auditory system is that the speech signal is pre-processed by certain sophisticated psychoacoustic mechanism. Psychoacoustics is the science that studies how human perceives sounds, including relationships between sound pressure level and loudness, human response to different frequencies and masking effects [18, 125]. In order to estimate the ‘new’ version of speech signal (after the pre-processing), in Chapter 3 we implement different kinds of masking effects sequentially and manage to obtain promising results. However, the sequential approach only deals with the ‘single-direction’ masking effect. For example, forward masking stands for the psychoacoustic effect that describes the relationship between the amount of masking and signal delay. Nevertheless, there is also masking effect between two speech signals that are different both in frequency and commencing time (temporal frequency masking).

In order to obtain a more complete psychoacoustic model, a novel 2D psychoacoustic filter is proposed. It implements temporal masking, lateral inhibition, temporal frequency masking, and temporal integration altogether with a simple 2D filter. Mathematical derivations are provided to show the correctness of the 2D psychoacoustic filter based on the characteristic functions of masking effects. Evaluation tests are carried out using the Aurora2 database.

4.2 2D Psychoacoustic Modeling

4.2.1 Temporal Masking

According to [2, 126], forward masking follows the following equation

$$M_{fm}(f, t, \Delta t) = a(f)[b(f) - \lg(\Delta t)][Y(f, t - \Delta t) - c(f)], \quad (4.1)$$

where f is the frequency index; t is the time index; $a(f)$, $b(f)$ and $c(f)$ are parameters listed in Table 4.1; M_{fm} is the amount of masking; Δt is the signal delay; $\lg(\cdot)$ is the common logarithm. Equation (4.1) describes how a speech signal, $Y(f, t - \Delta t)$, affects other acoustic signal that is later in time, $Y(f, t)$.

Since Cepstral Mean & Variance Normalization (CMVN) is utilized for all the algorithms in this thesis, the offset $c(f)$ is simply ignored in the following discussion. Equation (4.1) is further simplified to

$$M_{fm}(f, t, \Delta t) = a(f)[b(f) - \lg(\Delta t)]Y(f, t - \Delta t). \quad (4.2)$$

Define

$$A_{fm}(f, \Delta n) = a(f)[b(f) - \log(\Delta n \times T)]. \quad (4.3)$$

Table 4.1: Temporal Masking Parameters [2].

Frequency (Hz)	Parameters		
	$a(f)$	$b(f)$	$c(f)$
125	0.140	5.583	5.360
250	2.538	2.541	2.567
1000	2.877	2.886	2.890
4000	3.178	3.183	3.185

Equation (4.2) becomes

$$M_{fm}(f, t, \Delta t) = A_{fm}(f, \Delta n)Y(f, t - \Delta t). \quad (4.4)$$

Similarly, backward masking (BM) effect can be modeled as

$$M_{bm}(f, t, \Delta t) = A_{bm}(f, \Delta n)Y(f, t + \Delta t). \quad (4.5)$$

Figure 4.1: Sketch of psychoacoustic data.

Equations (4.4) and (4.5) describe how a speech signal affects other signal. However, in order to imitate how the human auditory system processes the speech signal, it is more important to study how a specific speech signal is affected by other signals

‘nearby’. For a sound element, $Y(f, t)$, in the time-frequency domain, theoretically, it should be affected by all the sound near it in time as is shown in Figure 4.1.

Therefore, the overall influence on the target element is a joint effect of all the elements ‘before’ and ‘after’ it in time,

$$M_{total}^{tm}(f, t) = \sum_{\Delta n=1}^{T_{fm}} A_{fm}(f, \Delta n)Y(f, t - \Delta n \times T) + \sum_{\Delta n=1}^{T_{bm}} A_{bm}(f, \Delta n)Y(f, t + \Delta n \times T). \quad (4.6)$$

Since backward masking is much weaker than forward masking, only forward masking is taken into consideration in the following discussion. Equation (4.6) becomes

$$M_{total}^{tm}(f, t) = \sum_{\Delta n=1}^{T_{fm}} A_{fm}(f, \Delta n)Y(f, t - \Delta n \times T). \quad (4.7)$$

After being processed by Temporal Masking, the speech signal becomes

$$\begin{aligned} \tilde{Y}(f, t) &= Y(f, t) - M_{total}^{tm}(f, t) \\ &= Y(f, t) - \sum_{\Delta n=1}^{T_{fm}} A_{fm}(f, \Delta n)Y(f, t - \Delta n \times T). \end{aligned} \quad (4.8)$$

Define the frame-wise speech vectors as

$$\begin{aligned} \mathbf{Y}_T(f) &= \begin{bmatrix} Y(f, 1) & Y(f, 2) & \dots & Y(f, T_s) \end{bmatrix} \\ \tilde{\mathbf{Y}}_T(f) &= \begin{bmatrix} \tilde{Y}(f, 1) & \tilde{Y}(f, 2) & \dots & \tilde{Y}(f, T_s) \end{bmatrix}, \end{aligned}$$

where T_s is the total frame number of the speech. Then, Equation (4.8) can be reformed as

$$\tilde{\mathbf{Y}}_T(f) = \mathbf{Y}_T(f) \otimes \mathbf{Mask}_{tm} \quad (4.9)$$

$$\mathbf{Mask}_{tm} = \begin{bmatrix} \mathbf{0}_{1 \times T_{fm}} & 1 & -A_{fm}(f, 1) & -A_{fm}(f, 2) & \dots & -A_{fm}(f, T_{fm}) \end{bmatrix},$$

where $\otimes$ stands for convolution; $\mathbf{0}_{1 \times T_{fm}}$ is $1 \times T_{fm}$ zero matrix; T_{fm} is the effective range of Temporal Masking effect; $\mathbf{Mask}_{tm}$ is the masker for Temporal Masking.

4.2.2 Simultaneous Masking

Simultaneous masking has been widely used in audio compression and speech enhancement [117]. However, its implementation in speech recognition is relatively new. Lateral Inhibition (LI) is one effective approach for implementing simultaneous masking [77, 106]. In neurobiology, lateral inhibition is used to measure the capacity of an excited neuron to reduce the activity of its neighbors. In speech processing, it describes how two speech signals with different frequencies occurring at the same time affect each other.

Following similar procedure as Temporal Masking, Lateral Inhibition can be modeled as

$$M_{total}^{li}(f, t) = \sum_{\Delta m \neq 0} A_{li}(\Delta m, t) Y(f + \Delta m \times F, t), \quad (4.10)$$

where Δm is the frequency bin difference defined in Equation (4.11); F is the frequency difference between neighboring frequency channels; $A_{li}(\Delta m, t)$ is the LI parameter from [77, 106].

$$\Delta m = \frac{\Delta f}{F}. \quad (4.11)$$

After being processed by Lateral Inhibition, the speech signal becomes

$$\begin{aligned} & \tilde{Y}(f, t) \\ = & Y(f, t) - M_{total}^{li}(f, t) \\ = & Y(f, t) - \sum_{-F_1 \leq \Delta m \leq F_2, \Delta m \neq 0} A_{li}(\Delta m, t) Y(f + \Delta m \times F, t), \end{aligned} \quad (4.12)$$

where F_1 and F_2 are parameters that are decided by the effective range of Simulta-

neous Masking. Define the frame speech vector as

$$\mathbf{Y}_F(t) = \begin{bmatrix} Y(1, t) \\ Y(2, t) \\ \vdots \\ Y(F_s, t) \end{bmatrix} \quad (4.13)$$

$$\tilde{\mathbf{Y}}_F(t) = \begin{bmatrix} \tilde{Y}(1, t) \\ \tilde{Y}(2, t) \\ \vdots \\ \tilde{Y}(F_s, t) \end{bmatrix}, \quad (4.14)$$

where F_s is the total frequency channel number of the speech. Equation (4.12) can be rearranged as

$$\tilde{\mathbf{Y}}_F(t) = \mathbf{Y}_F(t) \otimes \mathbf{Mask}_{\mathbf{i}}, \quad (4.15)$$

where $\mathbf{Mask}_{\mathbf{i}}$ is the masker for Simultaneous Masking defined as

$$\mathbf{Mask}_{\mathbf{i}} = \begin{bmatrix} -A_{li}(F_2) \\ -A_{li}(F_2 - 1) \\ \vdots \\ -A_{li}(1) \\ 1 \\ -A_{li}(-1) \\ \vdots \\ -A_{li}(-F_1 + 1) \\ -A_{li}(-F_1) \end{bmatrix}. \quad (4.16)$$

4.2.3 Overall Joint Masking Effects

In the time-frequency domain, speech components are influenced by all the surrounding components. It means for a speech signal $Y(f_i, t_i)$ it is affected by all

the other speech signals within a certain range, $\{Y(f_i + \Delta f_i, t_i + \Delta t_i) \mid 0 \leq \Delta t_i \leq T_{fm}, F_1 \leq \Delta f_i \leq F_2\}$. T_{fm} , $-F_1$, and F_2 are the effective ranges of Forward Masking and Lateral Inhibition.

Figure 4.2: Joint Masking Effect.

Figure 4.2 gives a better view of the above mentioned effect. All the speech components within the rectangle contributes to the final masking effect. The slashed area can be modeled using Temporal Masking defined in Equation (4.7) and the crossed area can be modeled by Frequency Masking defined in Equation (4.10). Finally, the remaining speech components in the rectangle can be modeled using the frequency dependent Forward Masking function from Jesteadt's paper [2], which will be discussed in detail in later parts.

The overall joint masking effect can be described as

$$\begin{aligned}
 & M_{total}(f_i, t_i) \\
 = & \sum_{\Delta t_i=1}^{T_{fm}} A_{fm}(f_i, \Delta t_i) Y(f_i, t_i - \Delta t_i) + \sum_{\Delta f_i \neq 0} A_{li}(\Delta f_i, t_i) Y(f_i - \Delta f_i, t_i) \\
 & + \sum_{\Delta f_i \neq 0} \sum_{\Delta t_i \neq 0} A_{diag}(\Delta f_i, \Delta t_i) Y(f_i - \Delta f_i, t_i - \Delta t_i). \tag{4.17}
 \end{aligned}$$

To get a unified form of 2D psychoacoustic filter, a new masking parameter is defined as

$$\alpha(\Delta f_i, \Delta t_i) = \begin{cases} 1 & \Delta f_i = 0, \Delta t_i = 0 \\ A_{fm}(f_i, \Delta t_i) & \Delta f_i = 0, \Delta t_i \neq 0 \\ A_{li}(\Delta f_i, t_i) & \Delta f_i \neq 0, \Delta t_i = 0 \\ A_{diag}(\Delta f_i, \Delta t_i) & \Delta f_i \neq 0, \Delta t_i \neq 0 \end{cases}. \quad (4.18)$$

Then, the total masking effect becomes

$$\begin{aligned} & M_{total} \quad (4.19) \\ & = \sum_{\Delta t_i=1}^T \sum_{\Delta f_i=-F_1}^{F_2} \alpha(\Delta f_i, \Delta t_i) Y(f_i - \Delta f_i, t_i - \Delta t_i) - \alpha(0, 0) Y(f_i, t_i), \end{aligned}$$

where $\alpha(\Delta f_i, \Delta t_i)$ is the filter parameter, and $F_1 > F_2$ according to the effective range of masking effects.

Equation (4.19) gives the total masking effect introduced by the surrounding speech components. As masking effects tend to weaken speech, Equation (4.19) describes how surrounding speech components weaken the centre component. Therefore, the target speech component changes to

$$\begin{aligned} & \tilde{Y}(f_i, t_i) \\ & = Y(f_i, t_i) - M_{total} \quad (4.20) \\ & = \sum_{\Delta t_i=1}^{T_{fm}} \sum_{\Delta f_i=F_1}^{F_2} [-\alpha(\Delta f_i, \Delta t_i)] Y(f - \Delta f_i, t - \Delta t_i). \end{aligned}$$

Define

$$\mathbf{Y}_{\mathbf{FT}} = \begin{bmatrix} Y(1, 1) & Y(2, 1) & \cdots & Y(1, T_s) \\ Y(2, 1) & Y(2, 2) & \cdots & Y(2, T_s) \\ \vdots & \vdots & \ddots & \vdots \\ Y(F_s, 1) & Y(F_s, 2) & \cdots & Y(F_s, T_s) \end{bmatrix} \quad (4.21)$$

$$\tilde{\mathbf{Y}}_{\mathbf{FT}} = \begin{bmatrix} \tilde{Y}(1,1) & \tilde{Y}(2,1) & \cdots & \tilde{Y}(1,T_s) \\ \tilde{Y}(2,1) & \tilde{Y}(2,2) & \cdots & \tilde{Y}(2,T_s) \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{Y}(F_s,1) & \tilde{Y}(F_s,2) & \cdots & \tilde{Y}(F_s,T_s) \end{bmatrix}. \quad (4.22)$$

Then

$$\tilde{\mathbf{Y}}_{\mathbf{FT}} = \mathbf{Y}_{\mathbf{FT}} * \mathbf{Mask}, \quad (4.23)$$

where

$$\mathbf{Mask} = \begin{bmatrix} \mathbf{0}_{(F_1-F_2) \times T_{fm}} & & \mathbf{0}_{(F_1-F_2) \times (T_{fm}+1)} & & \\ & -\alpha(F_2, 0) & & & -\alpha(F_2, -T_{fm}) \\ & \vdots & & \ddots & \\ & -\alpha(1, 0) & -\alpha(1, -1) & & \\ \mathbf{0}_{(F_1+F_2+1) \times T_{fm}} & 1 & -\alpha(0, -1) & \cdots & -\alpha(0, -T_{fm}) \\ & -\alpha(-1, 0) & -\alpha(-1, -1) & & \\ & \vdots & & \ddots & \\ & -\alpha(-F_1, 0) & & & -\alpha(F_1, -T_{fm}) \end{bmatrix}. \quad (4.24)$$

4.2.4 Temporal Integration

Masking effect mainly describes how the amount of masking is affected by time and frequency. However, further study shows that the duration of speech signal also greatly affects the total amount of masking, which is the so called Temporal Integration (TI). According to [1, 126], when signal duration increases, there is a great decrease in the mean masking thresholds (or the amount of masking). For example, in the experimental data presented by Oxenham et al. shown in Figure 4.3 ([1]: Fig 1, pp735), at an offset of 9 ms, mean thresholds decreased by nearly

14 dB as the signal duration increased from 2 to 7 ms. In other words, in [1, 126], an increase of 5 ms (7 ms - 2 ms) resulted in a 14 dB decrease in the amount of masking. Note that at the duration of 2 ms, the amount of masking is about 56 dB. It can be found out that the amount of masking drops about 25% due to a relatively slight increase (5 ms) in the signal duration.

Figure 4.3: TI experimental results [1].

As is known to all, since speech has active/non-active periods, its power is more concentrated in certain areas, both stronger in energy and longer in duration. Therefore, temporal integration tends to impose more influence on the speech. Then the total amount of masking becomes,

$$\tilde{M}_{total} = \begin{cases} M_{total} & , nonspeech \\ M_{total} - M_{TI} & , speech \end{cases} \quad (4.25)$$

where M_{TI} stands for the decrease of masking caused by temporal integration.

After being processed by the new 2D filter, the speech power spectrum becomes

$$\begin{aligned}
& |\tilde{Y}(f_i, t_i)|^2 \\
&= |Y(f_i, t_i)|^2 * \tilde{Mask}_{\Delta f_i, \Delta t_i} \\
&= (|X(f_i, t_i)|^2 + |\tilde{N}(f_i, t_i)|^2) * \tilde{Mask}_{\Delta f_i, \Delta t_i} \\
&= |X(f_i, t_i)|^2 * (Mask_{\Delta f_i, \Delta t_i} + Mask_{TI}) + |N(f_i, t_i)|^2 * Mask_{\Delta f_i, \Delta t_i}.
\end{aligned} \tag{4.26}$$

Then, the clean speech becomes $|X(f_i, t_i)|^2 * (Mask_{\Delta f_i, \Delta t_i} + Mask_{TI})$, which will be denoted as $|\tilde{X}(f_i, t_i)|^2$. Therefore,

$$\begin{aligned}
& \frac{|\tilde{X}(f_i, t_i)|^2}{|\tilde{Y}(f_i, t_i)|^2} \\
&= \frac{|X(f_i, t_i)|^2 * (Mask_{\Delta f_i, \Delta t_i} + Mask_{TI})}{|X(f_i, t_i)|^2 * (Mask_{\Delta f_i, \Delta t_i} + Mask_{TI}) + |N(f_i, t_i)|^2 * Mask_{\Delta f_i, \Delta t_i}} \\
&> \frac{|X(f_i, t_i)|^2}{|X(f_i, t_i)|^2 + |N(f_i, t_i)|^2}.
\end{aligned} \tag{4.27}$$

From Equation (4.27), it can be easily seen that temporal integration can help to increase the SNR of the noisy speech. In our present implementation, Temporal Integration is calculated by

$$M_{TI} = \alpha_{TI} Y(f_i, t_i), \tag{4.28}$$

where α_{TI} is the parameter for calculating TI.

Equation (4.20) becomes

$$\tilde{Y}(f_i, t_i) = \begin{cases} Y(f_i, t_i) - M_{total} & , nonspeech \\ Y(f_i, t_i) - M_{total} + M_{TI} & , speech \end{cases}. \tag{4.29}$$

Then the 2D psychoacoustic filter becomes

$$\mathbf{Mask} = \begin{bmatrix} \mathbf{0}_{(F_1-F_2) \times T_{fm}} & \mathbf{0}_{(F_2-F_1) \times (T_{fm}+1)} \\ \mathbf{0}_{(F_1+F_2+1) \times T_{fm}} & \hat{M} \end{bmatrix} \tag{4.30}$$

where $\hat{\mathbf{M}}(f)$ is defined as

$$\hat{\mathbf{M}} = \begin{bmatrix} \mathbf{0}_{(F_1-F_2) \times T_{fm}} & \mathbf{0}_{(F_2-F_1) \times (T_{fm}+1)} & & & \\ & -\alpha(F_2, 0) & & & -\alpha(F_2, -T_{fm}) \\ & \vdots & & \ddots & \\ & -\alpha(1, 0) & -\alpha(1, -1) & & \\ \mathbf{0}_{(F_1+F_2+1) \times T_{fm}} & 1 + \alpha_{TI} & -\alpha(0, -1) & \cdots & -\alpha(0, -T_{fm}) \\ & -\alpha(-1, 0) & -\alpha(-1, -1) & & \\ & \vdots & & \ddots & \\ & -\alpha(-F_1, 0) & & & -\alpha(-F_1, -T_{fm}) \end{bmatrix}. \quad (4.31)$$

The proposed 2D psychoacoustic filter enhances the high frequencies and helps to sharpen the spectral peaks so as to improve the performance of the recognition system. For simplicity, $\hat{M}$ will be referred to as the 2D psychoacoustic filter in the following discussion.

4.3 Parameter Design

4.3.1 Original 2D Filter

The proposed 2D psychoacoustic filter is designed to achieve the effects of both Lateral Inhibition (LI) and Temporal Masking (TM) at the same time. Parameters of the 2D psychoacoustic modeling are developed based on the 1D Mexican-hat, i.e. $[-0.07 \quad -0.27 \quad -0.16 \quad 1 \quad -0.16 \quad -0.27 \quad -0.07]$. Figure 4.4 shows the 1D Mexican Hat.

By assuming that lateral inhibition and temporal masking share the same set of

Figure 4.4: 1D Mexican Hat.

parameters [48, 102], the initial parameter set of 2D filter is shown in Table 4.2

Table 4.2: Initial Parameter Set of 2D Filter.

Freq\Time ($f \setminus t$)	-3	-2	-1	0	1	2	3
-3				-0.07			
-2				-0.27			
-1				-0.16			
0	-0.07	-0.27	-0.16	1	-0.16	-0.27	-0.07
1				-0.16			
2				-0.27			
3				-0.07			

The parameters in diagonal directions are calculated from Table 4.2 using linear interpolation.

The distance between the diagonal coefficients, $mask(f, t)$, and the centre coefficient, $mask(0, 0)$, is denoted as d . d can be calculated using the following equation,

$$d = \sqrt{t^2 + f^2} \quad (4.32)$$

As the masker is symmetric, the calculation can be performed using just one side of the curve shown in Figure 4.4. Take the right side of the curve as an example.

The function of the characteristic curve is

$$mask(d) = \begin{cases} 1 - 1.16d & , \quad 0 \leq d < 1 \\ -0.05 - 0.11d & , \quad 1 \leq d < 2 \\ -0.67 + 0.2d & , \quad 2 \leq d < 3 \\ -0.07 + \frac{0.07}{3\sqrt{2}-3}(d-3) & , \quad 3 \leq d \leq 3\sqrt{2} \end{cases} \quad (4.33)$$

It has to be noted that the last section is $3 \leq x \leq 3\sqrt{2}$ making the total input of Equation (4.33) to $-3\sqrt{2} \leq x \leq 3\sqrt{2}$. The objective is to have four zeros at the corners of the 2D mask, $mask(\pm 3, \pm 3)$, shown in Table 4.3. The diagonal coefficient value is a function of d and it can be obtained from Table 4.2 using linear interpolation as described by Equation (4.33). Heuristically, 40 is an optimal value for the centre element.

Table 4.3: The Original 2D Filter.

Freq\Time ($f \setminus t$)	-3	-2	-1	0	1	2	3
-3	0	-0.0359	-0.0609	-0.07	-0.0609	-0.0359	0
-2	-0.0359	-0.1043	-0.2228	-0.27	-0.2228	-0.1043	-0.0359
-1	-0.0609	-0.2228	-0.2056	-0.16	-0.2056	-0.2228	-0.0609
0	-0.07	-0.27	-0.16	40	-0.16	-0.27	-0.07
1	-0.0609	-0.2228	-0.2056	-0.16	-0.2056	-0.2228	-0.0609
2	-0.0359	-0.1043	-0.2228	-0.27	-0.2228	-0.1043	-0.0359
3	0	-0.0359	-0.0609	-0.07	-0.0609	-0.0359	0

4.3.2 Temporal Warped (TW) 2D Psychoacoustic Filter

The original 2D filter is developed based on the assumption that lateral inhibition and temporal masking share the same set of parameters. However, further study shows that the validity of the assumption depends on many things such as sampling rate, frame rate and so on. Therefore, temporal warping is adopted to improve the performance of the 2D psychoacoustic filter.

There are different mathematical models for describing temporal masking effect. All of them [107, 116, 127] come to a common conclusion that forward masking is more effective than backward masking. As shown in Figure 3.1, the effective range of forward masking is much longer than that of backward masking. Thus the parameters of temporal masking cannot be symmetric. It has to be warped to get a better set of temporal masking parameters. Since the 1D Mexican Hat is widely used as lateral inhibition parameters [102, 116], it is chosen to be the primary parameter for temporal masking since it follows the general tendency of masking effects. Each side of the mask is linearly warped. Figure 4.5 shows the warped mask. The detailed steps for calculation are given in the following section.

Figure 4.5: Warped Mexican Hat.

Starting from the 1D Mexican hat in Figure 4.4, the warping process consists of two steps. Firstly, each side of the Mexican hat is modified proportionally, making the right side $7/4$ of the original length and the left side $1/4$ of the original length. Figure 4.5 shows the warped parameter and the original. Each side of the Mexican hat contains four sections (right side: $[0, 1)$, $[1, 2)$, $[2, 3)$, $[3, 4)$; left side: $[-1, 0)$, $[-2, -1)$, $[-3, -2)$, $[-4, -3)$). The length of each interval is 1. After the first step, the right side becomes $[0, 1.75)$, $[1.75, 3.5)$, $[3.5, 5.25)$, $[5.25, 7)$, and the left

side becomes $[-0.25, 0)$, $[-0.5, -0.25)$, $[-0.75, -0.5)$, $[-1, -0.75)$. The length of the intervals on the left side is 0.25 and the interval length of the right side is 1.75. Like Equation (4.33), the two sections at the end ($[5.25, 7]$ and $[-1, -0.75]$) need further modification. In Equation (4.33), the last interval is extended to $3\sqrt{2} - 3$ times the original length. Similar operation is made to the warped Mexican hat. The second step is to extend interval $[5.25, 7]$ and $[-1, -0.75]$ to $[5.25, 5.25 + 1.75 \times \frac{3\sqrt{2}-3}{1}]$ and $[-0.75 - 0.25 \times \frac{3\sqrt{2}-3}{1}, -0.75]$ respectively. Hence, the linear interpolation equation for the warped Mexican hat is

$$mask(d) = \begin{cases} -0239 - 02253d & , \quad -1.06 \leq d \leq -0.75 \\ -067 - 08d & , \quad -0.75 < d \leq 0.5 \\ -005 + 044d & , \quad -0.5 < d \leq -0.25 \\ 1 + 4.64d & , \quad -0.25 < d < 0 \\ 1 - 0.6629d & , \quad 0 \leq d < 1.75 \\ -005 - 00629d & , \quad 1.75 \leq d < 3.5 \\ -067 + 01143d & , \quad 3.5 \leq d < 5.25 \\ -0239 + 00322d & , \quad 5.25 \leq d \leq 7.42 \end{cases} \quad (4.34)$$

The warped parameters are calculated using Equation (4.34) by setting $d = -1, 0, 1, 2, 3, 4, 5$. The warped parameters are as follows,

$$[-0.0137 \quad 1 \quad 0.3371 \quad -0.1757 \quad -0.2386 \quad -0.2129 \quad -0.0986].$$

Following a similar procedure as the original 2D filter, the initial parameter set of the warped 2D filter is shown in Table 4.4.

In order to develop an equation to calculate the diagonal parameters, two things have to be made clear. Firstly, the proposed filter is implemented by convolution in the time-frequency domain. Another thing that has to be mentioned is the physical

Table 4.4: Initial Parameter Set of the Temporal Warped 2D Filter.

Freq\Time ($f \setminus t$)	-1	0	1	2	3	4	5
-3		-0.07					
-2		-0.27					
-1		-0.16					
0	-0.0137	1	0.3371	-0.1757	-0.2386	-0.2129	-0.0986
1		-0.16					
2		-0.27					
3		-0.07					

meaning of the parameters. The centre element stands for the target speech element. According to temporal masking and simultaneous masking, in time-frequency domain all the neighboring sound weakens the target sound. Therefore, all the other 48 elements of the mask are used to measure how much the signal at that location weakens the target signal. For that reason except the centre element all the parameters should be negative, which accounts for the weakening effect. Besides, masking effect will become weaker as two sounds become farther from each other. Therefore, the equation for diagonal parameters should possess the following form,

$$mask(t, f) = \begin{cases} 1 & , f = 0, t = 0 \\ -\frac{[x^2(t) + y^2(f)]^{1/2}}{(t^2 + f^2)^{1/2}} & , \text{others} \end{cases} \quad (4.35)$$

where $x(t) = \{-0.0800, 1, 0.3371, -0.1757, -0.2386, -0.2129, -0.0986\}$, $t = -1, 0, \dots, 4, 5$; $y(f) = \{-0.07, -0.27, -0.16, 1, -0.16, -0.27, -0.07\}$, $f = -3, -2, \dots, 2, 3$; f stands for the frequency index of the mask, and t refers to the time index of the mask; $x(f)$ and $y(f)$ are the basic parameters; the minus sign corresponds to the fact that all the surrounding speech signal in the time-frequency domain tend to weaken the centre element, $mask(0, 0)$; $1 / (t^2 + f^2)^{1/2}$ corresponds to the fact that masking effect becomes weaker as two sounds become farther from each other. With the initial parameters from Table 4.4, parameters of the mask are

Table 4.5: Temporal Warped 2D Psychoacoustic Filter.

Freq\Time ($f \setminus t$)	-1	0	1	2	3	4	5
-3	-0.0226	-0.3341	-0.1089	-0.0525	-0.0586	-0.0448	-0.0207
-2	-0.1209	-0.5179	-0.1932	-0.1139	-0.0999	-0.0769	-0.0534
-1	-0.1136	-1.0127	-0.2639	-0.1063	-0.0908	-0.0646	-0.0369
0	-1.0001	40	-1.0553	-0.5077	-0.3427	-0.2556	-0.201
1	-0.1136	-1.0127	-0.2639	-0.1063	-0.0908	-0.0646	-0.0369
2	-0.1209	-0.5179	-0.1932	-0.1139	-0.0999	-0.0769	-0.0534
3	-0.0226	-0.3341	-0.1089	-0.0525	-0.0586	-0.0448	-0.0207

calculated using Equation (4.35). Following the same procedure discussed in Section 4.3.1, the centre element is changed to 40. The warped mask is shown in Table 4.5.

4.3.3 Temporal Frequency Warped (TFW) 2D Psychoacoustic Filter

Figure 4.6 shows the characteristic curve of forward masking which describes how the amount of masking changes with time [2].

The sampling frequency of the AURORA2 database is 8 kHz. The frame length is chosen to be 200 (according to the Aurora2 database demo scripts). After transformation into the time-frequency domain, the frequency channels that are covered are 40 Hz $\sim$ 4000 Hz. Theoretically, forward masking parameters are different for different frequency. However, strictly following the above mentioned property will make the proposed filter too complicated for practical implementation. Therefore, the speech is split into two bands, high band and low band. The parameters are chosen to be fixed within different frequency bands. In this thesis, the 4 kHz and 1 kHz parameter sets are adopted for further analysis. For lateral inhibition, the

Figure 4.6: Psychoacoustic Data of FM.

Table 4.6: Design of LI Parameters.

Freq. Index	-3	-2	-1	0	1	2	3	4	5
LI Filter	-0.07	-0.27	-0.16	1	-0.16	-0.27	-0.07		
Masking Para.	0.07	0.27	0.16	0	0.16	0.27	0.07		
Warped Para.			0.0137	0	0.0914	0.1757	0.2386	0.2129	0.0986
Warped LI Para.			-0.0137	1	-0.0914	-0.1757	-0.2386	-0.2129	-0.0986

warped parameters from Section 4.3.2 are adopted for further study. The detailed parameters are given in Table 4.6.

The diagonal parameters are one of the most important characteristics of the proposed algorithm. Most of the current algorithms focus on the relationship between two parameters. For example, as shown in Figure 4.7, forward masking focuses on how the amount of masking changes with time, and lateral inhibition studies the relationship between the amount of masking and frequency. However, it has to be noted that two signals that are different in both time and frequency also possess masking effect.

A two-step process is developed to calculate the amount of masking between two

Figure 4.7: Masking Effect.

speech signals of frequencies f_1 and f_2 with time difference Δt . Firstly, the forward masking result for Δt is calculated using FM characteristic function. Then the amount of masking from FM is used to generate the LI result. Equation (4.36) gives a better view of the above mentioned process. Similarly, the diagonal result can also be calculated by the LI-first process shown in Equation (4.37) and illustrated by Route 2 in Figure 4.8.

Route 1:

$$M^{diag}(f, t) = F_{li}\{F_{fm}[Y(f, t), \Delta t], \Delta f\} \quad (4.36)$$

Route 2:

$$M^{diag}(f, t) = F_{fm}\{F_{li}[Y(f, t), \Delta f], \Delta t\} \quad (4.37)$$

Equations (4.36) and (4.37) can be viewed as going along two different routes in the time-frequency domain. As is shown in Figure 4.8, route 1 corresponds to Equation (4.36) while route 2 shows Equation (4.37). The diagonal parameter can

Figure 4.8: Design of 2D Psychoacoustic Filter.

be calculated by

$$M^{diag}(f, t) = F_{fm}\{F_{li}[Y(f, t)]\} = F_{li}\{F_{fm}[Y(f, t)]\} = \alpha_{fm}\alpha_{li}Y(f, t). \quad (4.38)$$

Then taking into account the physical meaning of the proposed 2D filter

$$\mathbf{M}^{diag}(f, t) = -\alpha_{fm}\alpha_{li}Y(f, t), \quad (4.39)$$

where α_{fm} and α_{li} stand for FM and LI parameters respectively

It has to be noted that under the above mentioned assumption, Route 1 and Route 2 should achieve the same results.

$$F_{fm}\{F_{li}[Y(f, t)]\} = F_{li}\{F_{fm}[Y(f, t)]\}, \quad (4.40)$$

where F_{fm} stands for the FM characteristic function and F_{li} is the LI characteristic function.

FM and LI sharing similar form in the characteristic function is the sufficient condition for Equation (4.40)'s validity. It is well known that FM and LI share

similar property [2, 116, 120]. Based on the above mentioned fact, the two-step calculation approach for diagonal parameters is very reasonable since it follows the basic properties of psychoacoustics. The final form of the temporal frequency warped 2D psychoacoustic filter is given in Table 4.7.

Table 4.7: Temporal Frequency Warped 2D Psychoacoustic Filter.

Freq\Time	0	1	2	3	4	5	6
-1	-0.0137	-0.0065	-0.005	-0.0041	-0.0034	-0.0029	-0.0025
0	5	-0.4736	-0.3622	-0.2971	-0.2508	-0.215	-0.1857
1	-0.0914	-0.0433	-0.0331	-0.0272	-0.0229	-0.0196	-0.017
2	-0.1757	-0.0832	-0.0636	-0.0522	-0.0441	-0.0378	-0.0326
3	-0.2386	-0.113	-0.0864	-0.0709	-0.0598	-0.0513	-0.0443
4	-0.2129	-0.1008	-0.0771	-0.0632	-0.0534	-0.0458	-0.0395
5	-0.0986	-0.0467	-0.0357	-0.0293	-0.0247	-0.0212	-0.0183
Freq\Time	7	8	9	10	11	12	13
-1	-0.0022	-0.0019	-0.0017	-0.0014	-0.0012	-0.001	-0.0008
0	-0.1609	-0.1395	-0.1205	-0.1036	-0.0883	-0.0743	-0.0614
1	-0.0147	-0.0127	-0.011	-0.0095	-0.0081	-0.0068	-0.0056
2	-0.0283	-0.0245	-0.0212	-0.0182	-0.0155	-0.0131	-0.0108
3	-0.0384	-0.0333	-0.0288	-0.0247	-0.0211	-0.0177	-0.0147
4	-0.0343	-0.0297	-0.0257	-0.0221	-0.0188	-0.0158	-0.0131
5	-0.0159	-0.0138	-0.0119	-0.0102	-0.0087	-0.0073	-0.0061
Freq\Time	14	15	16				
-1	-0.0007	-0.0005	-0.0004				
0	-0.0495	-0.0384	-0.0281				
1	-0.0045	-0.0035	-0.0026				
2	-0.0087	-0.0068	-0.0049				
3	-0.0118	-0.0092	-0.0067				
4	-0.0105	-0.0082	-0.006				
5	-0.0049	-0.0038	-0.0028				

4.3.4 Adaptive 2D Psychoacoustic Filter

As the parameters of masking effects change with frequency, ideally there should be different 2D psychoacoustic filter for different frequency. For a speech sample denoted as $F_s \times T_s$ matrix, it is divided equally into two different parts, low band

and high band.

$$\mathbf{Y}_s = \begin{bmatrix} \mathbf{Y}_{s1} \\ \mathbf{Y}_{s2} \end{bmatrix}, \quad (4.41)$$

where $\mathbf{Y}_{s1}$ and $\mathbf{Y}_{s2}$ are defined as

$$\mathbf{Y}_{s1} = \begin{bmatrix} Y(1, 1) & Y(1, 2) & \cdots & Y(1, T_s) \\ \vdots & \vdots & & \vdots \\ Y(\frac{1}{2}F_s, 1) & Y(\frac{1}{2}F_s + 1, 2) & \cdots & Y(\frac{1}{2}F_s + 1, T_s) \end{bmatrix} \quad (4.42)$$

$$\mathbf{Y}_{s2} = \begin{bmatrix} Y(\frac{1}{2}F_s + 1, 1) & Y(\frac{1}{2}F_s + 1, 2) & \cdots & Y(\frac{1}{2}F_s + 1, T_s) \\ \vdots & \vdots & & \vdots \\ Y(F_s, 1) & Y(F_s, 2) & \cdots & Y(F_s, T_s) \end{bmatrix}. \quad (4.43)$$

Each band is processed by different 2D psychoacoustic filter. The 1 kHz and 4 kHz parameters in Table 4.1 are used for low band and high band temporal masking parameters, respectively.

For the implementation of Temporal Integration (TI), the centre parameter should be different between speech and non-speech frames. The TI parameter, α_{TI} , is obtained empirically. The optimal value is given in Table 4.8.

Table 4.8: Temporal Integration Parameter.

	Speech	Non-speech
Low Band	4	3
High Band	3	2

Figure 4.12 illustrates the proposed algorithm. After DFT, the speech spectrogram is divided into high band and low band. A voice activity detector (VAD) is utilized to distinguish speech/non-speech frames.

For each band, two different Temporal Integration parameters are used. Therefore, there are overall four different 2D psychoacoustic filters used for implementa-

Figure 4.9: Block diagram of adaptive 2D psychoacoustic filtering.

tion. As is shown in Figure 4.10, four different maskers are adopted for different situations.

Figure 4.10: Adaptive 2D Psychoacoustic Filtering.

In our present implementation, the noise is estimated using a minimum-controlled recursive moving-average noise tracker similar to the one described in [45, 128]. Generally, a decision on whether a frame contains speech or noise is made based on the energy ratio test [128]

$$\frac{|P_y(f, t)|_t^2}{|P_n(f, t)|_{\min}^2} > \nu, \quad (4.44)$$

where ν is the threshold, $|P_n(f, t)|_{\min}^2$ is the smoothed minimum noise power within

a sliding window which can be tracked efficiently and $|P_y(f, t)|_t^2$ is the smoothed (using adjacent channels) power of the noisy speech [45].

Since CMVN is implemented in the proposed algorithm, the scaling problem can be solved by using CMVN. Therefore, in our present implementation, the 2D psychoacoustic filter is not normalized.

Table 4.9: Temporal Frequency Warped 2D Psychoacoustic Filter (low band).

Freq\Time	0	1	2	3	4	5	6
-1	-0.0137	-0.0065	-0.005	-0.0041	-0.0034	-0.0029	-0.0025
0	$1 + \alpha_{TI}^{low}$	-0.4736	-0.3622	-0.2971	-0.2508	-0.215	-0.1857
1	-0.0914	-0.0433	-0.0331	-0.0272	-0.0229	-0.0196	-0.017
2	-0.1757	-0.0832	-0.0636	-0.0522	-0.0441	-0.0378	-0.0326
3	-0.2386	-0.113	-0.0864	-0.0709	-0.0598	-0.0513	-0.0443
4	-0.2129	-0.1008	-0.0771	-0.0632	-0.0534	-0.0458	-0.0395
5	-0.0986	-0.0467	-0.0357	-0.0293	-0.0247	-0.0212	-0.0183
Freq\Time	7	8	9	10	11	12	13
-1	-0.0022	-0.0019	-0.0017	-0.0014	-0.0012	-0.001	-0.0008
0	-0.1609	-0.1395	-0.1205	-0.1036	-0.0883	-0.0743	-0.0614
1	-0.0147	-0.0127	-0.011	-0.0095	-0.0081	-0.0068	-0.0056
2	-0.0283	-0.0245	-0.0212	-0.0182	-0.0155	-0.0131	-0.0108
3	-0.0384	-0.0333	-0.0288	-0.0247	-0.0211	-0.0177	-0.0147
4	-0.0343	-0.0297	-0.0257	-0.0221	-0.0188	-0.0158	-0.0131
5	-0.0159	-0.0138	-0.0119	-0.0102	-0.0087	-0.0073	-0.0061
Freq\Time	14	15	16				
-1	-0.0007	-0.0005	-0.0004				
0	-0.0495	-0.0384	-0.0281				
1	-0.0045	-0.0035	-0.0026				
2	-0.0087	-0.0068	-0.0049				
3	-0.0118	-0.0092	-0.0067				
4	-0.0105	-0.0082	-0.006				
5	-0.0049	-0.0038	-0.0028				

Table 4.9 gives the low band adaptive 2D psychoacoustic filter (without normalization). α_{TI}^{low} is defined as

$$\alpha_{TI}^{low} = \begin{cases} 4 & \text{Speech} \\ 3 & \text{Non - speech} \end{cases}. \quad (4.45)$$

The high band 2D psychoacoustic filter is given in Table 4.10. α_{TI}^{high} is defined as

$$\alpha_{TI}^{high} = \begin{cases} 3 & \text{Speech} \\ 2 & \text{Non - speech} \end{cases}. \quad (4.46)$$

Table 4.10: Temporal Frequency Warped 2D Psychoacoustic Filter (high band).

Freq\Time	0	1	2	3	4	5	6
-1	-0.0137	-0.0060	-0.0046	-0.0037	-0.0031	-0.0026	-0.0023
0	$1+\alpha_{TI}^{high}$	-0.4375	-0.3321	-0.2705	-0.2268	-0.1929	-0.1651
1	-0.0914	-0.0400	-0.0304	-0.0247	-0.0207	-0.0176	-0.0151
2	-0.1757	-0.0769	-0.0584	-0.0475	-0.0398	-0.0339	-0.0290
3	-0.2386	-0.1044	-0.0792	-0.0645	-0.0541	-0.0460	-0.0394
4	-0.2129	-0.0931	-0.0707	-0.0576	-0.0483	-0.0411	-0.0352
5	-0.0986	-0.0431	-0.0327	-0.0267	-0.0224	-0.0190	-0.0163
Freq\Time	7	8	9	10	11	12	13
-1	-0.0019	-0.0017	-0.0014	-0.0012	-0.001	-0.0008	-0.0007
0	-0.1417	-0.1214	-0.1035	-0.0875	-0.073	-0.0598	-0.0476
1	-0.013	-0.0111	-0.0095	-0.008	-0.0067	-0.0055	-0.0044
2	-0.0249	-0.0213	-0.0182	-0.0154	-0.0128	-0.0105	-0.0084
3	-0.0338	-0.0290	-0.0247	-0.0209	-0.0174	-0.0143	-0.0114
4	-0.0302	-0.0258	-0.0220	-0.0186	-0.0155	-0.0127	-0.0101
5	-0.014	-0.0120	-0.0102	-0.0086	-0.0072	-0.0059	-0.0047
Freq\Time	14	15	16				
-1	-0.0005	-0.0004	-0.0002				
0	-0.0364	-0.0259	-0.0161				
1	-0.0033	-0.0024	-0.0015				
2	-0.0064	-0.0045	-0.0028				
3	-0.0087	-0.0062	-0.0038				
4	-0.0077	-0.0055	-0.0034				
5	-0.0036	-0.0026	-0.0016				

4.4 Theoretical Analysis

4.4.1 Complex Spectral Processing

After being cut into frames and processed by Discrete Fourier Transform (DFT), the speech signal is transformed into the time-frequency domain,

$$Y(f, t) = Y_r(f, t) + i \times Y_i(f, t), \quad (4.47)$$

where i is the imaginary unit.

Normally, when extracting speech feature only power or magnitude spectrum is utilized. The phase information is just simply ignored. However, it has to be noted that there is useful information in the phase element of the speech [129, 130]. The proposed algorithm work directly in the time-frequency domain. In other words, the proposed algorithm makes use of the phase information in the noise removing process, i.e.

$$\begin{aligned} \tilde{Y}(f, t) &= Y(f, t) * Mask \\ &= [Y_r(f, t) + i \times Y_i(f, t)] * Mask \\ &= Y_r(f, t) * Mask + i \times Y_i(f, t) * Mask, \end{aligned} \quad (4.48)$$

where $*$ is the convolution operator.

4.4.2 Noise Removal

The general idea of the proposed algorithm is to simulate how the human auditory system processes speech and obtain the speech that is pre-processed by the human auditory system. From Equation (4.20), the pre-processed speech can be expressed as

$$\tilde{Y}(f, t) = Y(f, t) - M_{total}, \quad (4.49)$$

where M_{total} can be calculated as

$$M_{total} = \sum_{\Delta t=0}^{T_{fm}} \sum_{\Delta f=-F_1}^{F_2} \alpha(f, \Delta m, \Delta n) Y(f + \Delta m, t - \Delta n) - \alpha(f, 0, 0) Y(f, t). \quad (4.50)$$

This process is very similar to the Spectral Subtraction (SS) algorithm [38]. M_{total} can be treated as a kind of noise estimate. The noise estimate is generated based on a $(F_1 + F_2) \times T_{fm}$ window. State-of-the-art noise estimation algorithm usually adopts a 1D or 2D window to calculate the noise estimate. The difference lies in that the above mentioned algorithm generates the calculation using a weighted sum of all the speech elements in the $(F_1 + F_2) \times T_{fm}$ window. However, other noise estimation algorithms work by minimum statistics or simply averaging within a 1D or 2D window during the speech absent period.

Figure 4.11: Spectrogram of digit string '3Z82' from the AURORA2 database.

In order to give a better view of the proposed algorithm, an example is given to show the effectiveness of our proposed algorithm. Figure 4.11 shows the spectrogram of a sample speech before and after processed by the temporal frequency warped 2D psychoacoustic filter (given in Section 4.3.3). The sample speech is selected from the AURORA2 database. It can be easily observed that the proposed 2D psychoacoustic filter can effectively reduce noise.

4.5 Results and Discussion

4.5.1 Implementation Details

Evaluation tests are carried out using the AURORA2 database. Like LTFC (discussed in Chapter 3), the proposed 2D psychoacoustic filtering algorithm is developed based on the Mel Frequency Cepstral Coefficients (MFCC). The MFCC code from VoiceBox Toolkit [124] is adopted to be the baseline system. The baseline results are based on the standard 13 Mel Frequency Cepstral Coefficients (MFCC) together with the corresponding velocity and acceleration parameters. Hence there are totally 39 parameters, denoted as MFCC(39). The proposed front-end feature extractor is modified from the MFCC model by integrating a 2D psychoacoustic filter shown in Figure 4.12. In the following discussion, TW-2D refers to temporal warped 2D psychoacoustic filter given in Section 4.3.2. TFW-2D denotes the temporal frequency warped 2D psychoacoustic filter discussed in Section 4.3.3.

Figure 4.12: System Diagram of Adaptive 2D Psychoacoustic Filtering Algorithm.

4.5.2 Experimental Results

In this chapter, four different kinds of 2D psychoacoustic filters are given. They are presented in an evolutionary order. The Adaptive 2D psychoacoustic filter introduced

in Section 4.3.4 is the final form of the 2D psychoacoustic filter series. Detailed experimental results are given in Tables 4.11 and 4.12.

Table 4.11: Recognition Results of Adaptive 2D Filter (%) for Clean Training Condition.

	Noise Type	Clean	20	15	10	5	0	-5	Avg 0-20
Set A	Subway	99.42	97.73	95.98	92.60	84.03	64.26	29.66	86.92
	Babble	99.15	98.40	96.98	93.92	82.47	54.26	21.98	85.21
	Car	99.28	98.24	96.87	92.66	83.42	57.62	21.74	85.76
	Exhibition	99.72	97.72	94.63	89.20	76.55	54.92	28.48	82.60
Set B	Restaurant	99.42	98.68	97.27	93.43	83.21	58.09	27.11	85.86
	Street	99.15	97.76	96.34	92.53	83.04	59.64	25.85	86.14
	Airport	99.28	98.39	97.20	94.48	83.63	59.44	24.19	86.63
	Train	99.72	98.18	96.64	93.00	83.37	56.53	22.03	85.54
Set C	Restaurant	99.36	97.85	95.70	90.76	81.24	56.03	23.46	84.32
	Street	99.15	97.58	95.92	90.99	79.96	54.69	22.19	83.83
Avg		99.37	98.05	96.35	92.36	82.09	57.55	24.67	85.28

Table 4.12: Recognition Results of Adaptive 2D Filter (%) for Multi Training Condition.

	Noise Type	Clean	20	15	10	5	0	-5	Avg 0-20
Set A	Subway	98.86	98.40	97.76	96.47	93.92	81.49	54.22	93.61
	Babble	98.85	98.70	98.13	97.16	93.08	75.39	43.11	92.49
	Car	98.81	98.45	97.76	96.48	92.28	77.21	42.65	92.44
	Exhibition	99.14	98.55	97.59	95.06	89.26	74.58	49.49	91.01
Set B	Restaurant	98.86	99.05	98.56	97.42	92.42	76.88	45.87	92.87
	Street	98.85	98.43	97.58	95.83	91.32	77.24	46.25	92.08
	Airport	98.81	98.60	98.27	97.17	93.20	79.45	46.08	93.35
	Train	99.14	98.98	98.27	96.98	92.41	76.83	44.80	92.69
Set C	Restaurant	99.05	98.10	97.70	96.04	91.83	78.42	46.98	92.42
	Street	99.06	98.19	97.76	96.04	90.21	75.03	41.81	91.45
Avg		98.94	98.55	97.94	96.47	91.99	77.25	46.13	92.44

Comparison is mainly made against algorithms that are in relevant areas. That is Forward Masking (FM) [116], Lateral Inhibition (LI) [117], the Relative Spectra (RASTA) filter [79] as well as the Mean Variance Normalization & ARMA filtering (MVA) algorithm [47]. Experimental results are given in Tables 4.13 and 4.14. MFCC(39) stands for MFCC with velocity and acceleration components. All the

comparison methods are implemented based on MFCC(39). Experimental results are averaged over SNR of 0 dB to 20 dB denoted as Avg 0-20. 'Rel. Imp.' stands for relative improvements in terms of recognition rate.

Table 4.13: Recognition results for comparison targets under clean training condition (%).

SNR/dB	Clean	20	15	10	5	0	-5	Avg 0-20
MFCC(39)	99.36	97.37	93.51	81.16	56.02	28.39	13.04	71.29
FM	99.03	97.02	93.91	85.89	68.24	41.65	21.30	77.34
LI	99.42	97.19	94.23	83.29	60.92	34.21	17.07	73.97
CMVN	99.32	96.97	94.32	87.59	71.20	38.84	13.90	77.78
TW-2D	99.33	97.47	95.59	90.22	75.70	42.85	14.41	80.36
TFW-2D	99.38	97.77	95.94	91.61	80.42	56.26	24.37	84.40

Table 4.14: Recognition results for comparison targets under multi training condition (%).

SNR/dB	Clean	20	15	10	5	0	-5	Avg 0-20
MFCC(39)	99.11	98.18	97.60	95.52	87.61	60.37	26.83	87.85
FM	98.74	98.16	97.47	95.25	87.19	59.32	25.46	87.48
LI	99.13	98.19	97.62	95.53	88.06	61.93	26.59	88.26
CMVN	98.94	98.51	97.89	96.27	91.06	74.81	42.63	91.71
TW-2D	99.05	98.57	97.90	96.30	91.08	73.41	38.57	91.45
TFW-2D	98.87	98.35	97.80	96.09	91.09	75.73	43.86	91.81

The relative improvements in terms of Avg 0-20 are given in Tables 4.15 and 4.16.

Clean Training Condition

From the experimental results given in the above tables, it can be seen that the proposed algorithm show better results than all the comparison targets. Figure 4.13

Table 4.15: Relative Improvements under clean training condition (%).

SNR/dB	Clean	Avg 0-20	Rel. Imp	-5	Rel. Imp
MFCC(39)	99.36	71.29	19.62	13.04	90.03
FM	99.03	77.34	10.27	21.30	16.34
LI	99.42	73.97	15.29	17.07	45.17
CMVN	99.32	77.78	9.64	13.90	78.27
TW-2D	99.33	80.36	6.12	14.42	71.84
TFW-2D	99.38	84.40	1.04	24.37	1.68

Table 4.16: Relative Improvements under multi training condition (%).

SNR/dB	Clean	Avg 0-20	Rel. Imp	-5	Rel. Imp
MFCC(39)	99.11	87.85	5.22	26.83	71.93
FM	98.74	87.48	5.67	25.46	81.19
LI	99.13	88.26	4.73	26.59	73.49
CMVN	98.94	91.74	0.76	42.64	8.18
TW-2D	99.05	91.45	1.08	38.57	19.60
TFW-2D	98.67	91.81	0.69	43.86	5.18

gives a better view of the experimental results. Compared with MFCC(39), the proposed algorithm yields much better results. The relative improvement at Avg 0-20 is 19.62%, and at SNR of -5 dB it becomes 90.03%. For FM, LI and CMVN, the relative improvements at Avg 0-20 are 10.27%, 15.29%, and 9.64%. At SNR -5 dB, the relative improvements are 16.34%, 45.17%, and 78.27%, respectively.

In this chapter, three different 2D psychoacoustic filters are proposed, TW-2D filter, TFW-2D filter, and adaptive 2D psychoacoustic filter. Therefore, comparison is made among the proposed psychoacoustic filters. The relative improvements for TW-2D are 6.12% and 71.84% for Avg 0-20 and SNR -5 dB, respectively. For TFW-2D, the relative improvements are 1.04% and 1.68% for Avg 0-20 and SNR of -5 dB, respectively.

Figure 4.13: Experimental results (clean training condition).

Multi Training Condition

There are two training conditions in the AURORA2 database, clean and multi training conditions. For the multi training condition, since noisy speech is used to train HMMs, the recognition results are all very good. Therefore large improvements at this level are not very possible. However, the proposed algorithm still manages to get noticeable results. Figure 4.14 shows the relative improvements of the proposed algorithm over the comparison targets.

It can be seen that the proposed algorithm obtains very positive improvements. In terms of Avg 0-20, the relative improvements are 5.22% over MFCC(39), 5.67% over FM, 4.73% over LI, and 0.76% over CMVN. For SNR -5 dB, the relative improvements are 71.93% over MFCC, 81.19% over FM, 73.49% over LI, and 8.18% over CMVN. When compared with other 2D psychoacoustic filters, the Adaptive 2D filter manages to obtain good improvements. At Avg 0-20, the relative improvements are 1.08% over TW-2D and 0.69% over TFW-2D, respectively. For SNR of -5

Figure 4.14: Experimental results (multi training condition).

dB, the relative improvements are 19.60% over TW-2D and 5.18% over TFW-2D, respectively.

4.6 Conclusion

In this chapter, a hybrid feature extraction algorithm is proposed. It is successfully implemented in a MFCC based speech recognition system. The key feature of the proposed algorithm is that it successfully implements FM, LI and TI with a simple 2D psychoacoustic filter. It manages to reflect the asymmetrical nature of the human auditory system. Moreover, the proposed method does not need any additional training process, making the computational burden very low. Besides, due to the simplicity of the proposed algorithm it can be easily combined with other algorithm. Extensive comparison is made based on the AURORA2 database.

Chapter 5

2D Psychoacoustic L-Filter

5.1 Introduction

Psychoacoustics is the science that studies how the human auditory system works and the same theory can be applied to automatic speech recognition (ASR) systems. Undoubtedly, the human auditory system can work very well in different kinds of adverse environments, which will defeat most automatic speech recognition systems. Automatic speech recognition systems can achieve nearly perfect performance in a studio environment. However, even with a small amount of noise, e.g. SNR of 10 dB, the speech recognition accuracy may drop from 99% to 81% for MFCC feature (see Table 4.13 of previous chapter). As discussed in Chapters 3 and 4, speech signal is processed by the human auditory system and converted to neural spikes. Then the neural spikes are processed by the human brain. The idea of the psychoacoustic filters discussed in these chapters comes from the fact that a clearly audible sound (maskee) becomes weak or inaudible in the presence of another sound (masker). In other words, the maskee is weakened by the masker. Therefore, algorithms (see Chapters 3 and 4) are designed as the target speech signals (maskee), $Y(f_i, t_i)$,

minus the corresponding total amount of masking (calculated from the masker to show how much the masker weakens the maskee). Then we obtain the ‘new’ version (after pre-processed by the human auditory system) of the speech as

$$\tilde{Y}(f_i, t_i) = Y(f_i, t_i) - M_{total}. \quad (5.1)$$

For the human auditory system, any sound signal that is below the masking threshold cannot be heard. In other words, they are effectively equivalent for the human auditory system. For example, in Figure 5.1, Probes A and B (two weak sound pulses that are below the masking threshold) are the same for human listening and they are equivalent to zero-energy sound signal or silence. In the subtractive approach, we attempt to change all the weak signals (below the masking threshold) to zero and manage to obtain promising results. It strictly follows the masking effect. However, it has to be noted that the automatic speech recognition system is different from the human auditory system. The psychoacoustic models are optimized for the human auditory system. Therefore, it is reasonable to adapt the psychoacoustic models to automatic speech recognition systems.

Figure 5.1: Example of equivalent masking.

The subtractive implementation of psychoacoustic models changes all the weak speech signals below the threshold to zero, resulting in significant modification of

the speech. Although it can help to greatly remove noise, the subtractive approach also imposes distortion to the speech. There is always a compromise between the noise reduction ability and distortion. Therefore, it is straight forward to loosen the masking process or weaken the subtraction process. After processed by psychoacoustic models, the weak signal, Probe A in Figure 5.1, can be any value between zero and the masking threshold because they are effectively equivalent. Therefore, we modify Equation (5.1) to

$$\begin{aligned}
 & \tilde{Y}(f_i, t_i) \\
 = & Y(f_i, t_i) - \mu M_{total} \\
 = & Y(f_i, t_i) * \mathbf{Mask}_a.
 \end{aligned} \tag{5.2}$$

In this chapter, a new algorithm is proposed, which tries to change all the speech signals below the masking threshold to the optimal value (in terms of recognition rate) between the masking threshold and zero (or silence). Compared with the filters introduced in Chapter 4, all the filters introduced in this chapter are low-pass filters. Therefore, they will be referred to as L-filter and the algorithms discussed in Chapter 4 will be called H-filter in the following discussion. Theoretical analysis is given to show that the proposed L-filters can greatly remove noise and increase SNR.

5.2 Algorithm Description

5.2.1 Algorithm Description

In this part, we still use the frequency/time index (f_i and t_i) in our derivation (as defined in Chapter 4). Normally, the masking effects are implemented in a

subtractive way [48, 102]. According to Equation (4.20), the new version of speech after pre-processed by the human auditory system can be expressed as

$$\begin{aligned}
 & \tilde{Y}(f_i, t_i) \\
 = & Y(f_i, t_i) - M_{total} \\
 = & Y(f_i, t_i) * \mathbf{Mask},
 \end{aligned} \tag{5.3}$$

where $\mathbf{Mask}$ is the 2D psychoacoustic filter introduced in Chapter 4.

Equation (5.3) is modified to

$$\tilde{Y}(f_i, t_i) = Y(f_i, t_i) - \mu M_{total}. \tag{5.4}$$

Then,

$$\begin{aligned}
 & \tilde{Y}(f_i, t_i) \\
 = & Y(f_i, t_i) - \mu M_{total} \\
 = & Y(f_i, t_i) * \mathbf{Mask}_a,
 \end{aligned} \tag{5.5}$$

where $\mathbf{Mask}_a$ is defined as

$$\begin{aligned}
 & \mathbf{Mask}_a \\
 = & \mu \times \begin{bmatrix} \mathbf{0}_{(F_1-F_2) \times T_{fm}} & & \mathbf{0}_{(F_2-F_1) \times (T_{fm}+1)} & & \\ & -\alpha(F_2, 0) & & & -\alpha(F_2, -T_{fm}) \\ & \vdots & & \ddots & \\ & -\alpha(1, 0) & -\alpha(1, -1) & & \\ \mathbf{0}_{(F_1+F_2+1) \times T_{fm}} & 1/\mu + \alpha_{TI} & -\alpha(0, -1) & \cdots & -\alpha(0, -T_{fm}) \\ & -\alpha(-1, 0) & -\alpha(-1, -1) & & \\ & \vdots & & \ddots & \\ & -\alpha(-F_1, 0) & & & -\alpha(-F_1, -T_{fm}) \end{bmatrix}.
 \end{aligned} \tag{5.6}$$

The 2D filter given in Equation (5.5) possesses a very interesting property. If $\mu > 0$, it becomes the 2D psychoacoustic filter given in Chapter 4. If $\mu = 0$, the equation degrades to no masking. If $\mu < 0$, all the parameters in the filter become positive. It makes the filter becomes low-pass, which is the main topic of this chapter, i.e. L-filter. The physical meaning of Equation (5.4) is very clear. If the original speech signal is below the masking threshold, we reassign a small value (smaller than the masking threshold) to the signal based on the total amount of masking so as to optimize the final recognition rate.

Since cepstral mean and variance normalization (CMVN) is implemented in all the proposed algorithms, the modification (introduction of μ in Equation (5.4)) mainly affects the centre parameter of the filter (see Equation (5.8)). In the designing process of 2D psychoacoustic H-filters ($\mu > 0$), we've already optimized the centre component. For $\mu < 0$, heuristically -1 is the optimal value. Therefore, Equation (5.3) is modified to

$$\begin{aligned}
 & \tilde{Y}(f_i, t_i) \\
 = & Y(f_i, t_i) + M_{total} \\
 = & Y(f_i, t_i) * \mathbf{Mask}_{-1},
 \end{aligned} \tag{5.7}$$

where $\mathbf{Mask}_{-1}$ is defined as

$$\mathbf{Mask}_a = \begin{bmatrix} \mathbf{0}_{(F_1-F_2) \times T_{fm}} & & \mathbf{0}_{(F_2-F_1) \times (T_{fm}+1)} & & \\ & \alpha(F_2, 0) & & & \alpha(F_2, -T_{fm}) \\ & \vdots & & \ddots & \\ & \alpha(1, 0) & \alpha(1, -1) & & \\ \mathbf{0}_{(F_1+F_2+1) \times T_{fm}} & \alpha_{TI} - 1 & \alpha(0, -1) & \cdots & \alpha(0, -T_{fm}) \\ & \alpha(-1, 0) & \alpha(-1, -1) & & \\ & \vdots & & \ddots & \\ & \alpha(-F_1, 0) & & & \alpha(-F_1, -T_{fm}) \end{bmatrix}. \quad (5.8)$$

5.2.2 Theoretical Analysis

After processed by the proposed algorithm, the speech components become

$$\begin{aligned} & \tilde{Y}(f_i, t_i) \\ &= Y(f_i, t_i) + \tilde{M}_{total} \\ &= \xi \sum_{\Delta t_i=0}^{T_{fm}} \sum_{\Delta f_i=-F_1}^{F_2} \alpha(\Delta f_i, \Delta t_i) Y(f_i + \Delta f_i, t_i - \Delta t_i), \end{aligned} \quad (5.9)$$

where $\alpha_l(\Delta f_i, \Delta t_i)$ is the parameter given in Equation (5.8).

Since CMVN is implemented as part of the algorithm, Equation (5.9) becomes

$$\tilde{Y}(f_i, t_i) = \sum_{\Delta t_i=0}^{T_{fm}} \sum_{\Delta f_i=-F_1}^{F_2} \alpha(\Delta f_i, \Delta t_i) Y(f_i + \Delta f_i, t_i - \Delta t_i). \quad (5.10)$$

Speech is more concentrated in certain areas and completely absent in some other areas, which are normally called speech active periods and speech absent periods, respectively. Figure 5.2 gives an example of speech signal. In speech active periods, speech is more likely to possess stronger power than noise.

Figure 5.2: Sample speech signal.

After being processed by the above mentioned filter, the SNR becomes

$$\tilde{\gamma}_{snr} = \frac{|\tilde{X}(f_i, t_i)|^2}{|\tilde{N}(f_i, t_i)|^2} = \frac{|X(f_i, t_i)|^2 + M_{total}^s}{|N(f_i, t_i)|^2 + M_{total}^n}, \quad (5.11)$$

where M_{total}^s and M_{total}^n are the amount of masking for speech active period and speech absent period (noise period), respectively.

Forward masking is more dominant in the joint masking effects (see parameters in Tables 4.5 and 4.7), as a simple example, $M_{total}^s \approx \alpha_l(0, -1) [|X(f_i, t_i)|^2 + |N(f_i, t_i)|^2]$ and $M_{total}^n \approx \alpha_l(0, -1) |N(f_i, t_i)|^2$. Obviously, $M_{total}^s > M_{total}^n$.

Therefore,

$$\begin{aligned} & \tilde{\gamma}_{snr} \\ = & \frac{|X(f_i, t_i)|^2 + M_{total}^s}{|N(f_i, t_i)|^2 + M_{total}^n} \\ \approx & \frac{|X(f_i, t_i)|^2 + \alpha(0, -1) [|X(f_i, t_i)|^2 + |N(f_i, t_i)|^2]}{|N(f_i, t_i)|^2 + \alpha(0, -1) |N(f_i, t_i)|^2} \\ = & \frac{|X(f_i, t_i)|^2}{|N(f_i, t_i)|^2} + \frac{\alpha(0, -1)}{1 + \alpha(0, -1)} \\ > & \frac{|X(f_i, t_i)|^2}{|N(f_i, t_i)|^2}. \end{aligned} \quad (5.12)$$

The L-filter can help to greatly increase the SNR. It has to be noted that L-filter and H-filter are actually two different ways to increase SNR. SNR is defined as

$$\gamma_{snr} = \frac{|X(f_i, t_i)|^2}{|N(f_i, t_i)|^2}. \quad (5.13)$$

The H-filter works by

$$\tilde{\gamma}_{snr} = \frac{|X(f_i, t_i)|^2 - M_s}{|N(f_i, t_i)|^2 - M_n}, \quad (5.14)$$

where M_s and M_n are the amount of masking corresponding to speech and noise respectively.

For H-filter, the amount of masking corresponding to noise is nearly equal to the noise mean, $E(|N(f_i, t_i)|^2)$. Therefore, in Equation (5.14), the denominator is very small, which in return makes the SNR very high. When it comes to the L-filter, it mainly works by increasing the numerator.

5.3 2D L-filter design

By following the procedure introduced in Section 5.2.1, all the 2D psychoacoustic filters (H-filter) given in Chapter 4 have the corresponding L-filter form. The proposed psychoacoustic filter has the following form

$$\mathbf{Mask}_{\Delta \mathbf{f}, \Delta \mathbf{t}} = \begin{bmatrix} \mathbf{0}_{(F_1-F_2) \times T_{fm}} & \mathbf{0}_{(F_2-F_1) \times (T_{fm}+1)} \\ \mathbf{0}_{(F_1+F_2+1) \times T_{fm}} & \hat{M} \end{bmatrix}. \quad (5.15)$$

$\hat{M}$ is given in the following tables. Equation (5.16) and Table 5.1 define the low band Adaptive 2D L-filter.

$$\alpha_{TI}^{low} = \begin{cases} 3 & \text{Speech} \\ 2 & \text{Nonspeech} \end{cases}. \quad (5.16)$$

Table 5.1: Adaptive 2D Psychoacoustic L-Filter: low band.

Freq\Time	0	1	2	3	4	5	6
-1	0.0137	0.0065	0.005	0.0041	0.0034	0.0029	0.0025
0	$1 + \alpha_{TI}^{low}$	0.4736	0.3622	0.2971	0.2508	0.215	0.1857
1	0.0914	0.0433	0.0331	0.0272	0.0229	0.0196	0.017
2	0.1757	0.0832	0.0636	0.0522	0.0441	0.0378	0.0326
3	0.2386	0.113	0.0864	0.0709	0.0598	0.0513	0.0443
4	0.2129	0.1008	0.0771	0.0632	0.0534	0.0458	0.0395
5	0.0986	0.0467	0.0357	0.0293	0.0247	0.0212	0.0183
Freq\Time	7	8	9	10	11	12	13
-1	0.0022	0.0019	0.0017	0.0014	0.0012	0.001	0.0008
0	0.1609	0.1395	0.1205	0.1036	0.0883	0.0743	0.0614
1	0.0147	0.0127	0.011	0.0095	0.0081	0.0068	0.0056
2	0.0283	0.0245	0.0212	0.0182	0.0155	0.0131	0.0108
3	0.0384	0.0333	0.0288	0.0247	0.0211	0.0177	0.0147
4	0.0343	0.0297	0.0257	0.0221	0.0188	0.0158	0.0131
5	0.0159	0.0138	0.0119	0.0102	0.0087	0.0073	0.0061
Freq\Time	14	15	16				
-1	0.0007	0.0005	0.0004				
0	0.0495	0.0384	0.0281				
1	0.0045	0.0035	0.0026				
2	0.0087	0.0068	0.0049				
3	0.0118	0.0092	0.0067				
4	0.0105	0.0082	0.006				
5	0.0049	0.0038	0.0028				

Equation (5.17) and Table 5.2 define the high band Adaptive 2D L-filter

$$\alpha_{TI}^{high} = \begin{cases} 2 & \text{Speech} \\ 1 & \text{Nonspeech} \end{cases}. \quad (5.17)$$

5.4 Results and Discussion

5.4.1 Experimental Results

Evaluation tests are carried out using the AURORA2 database. The proposed L-2D psychoacoustic filters are developed based on the Mel Frequency Cepstral Co-

Table 5.2: Adaptive 2D Psychoacoustic L-Filter: high band.

Freq\Time	0	1	2	3	4	5	6
-1	0.0137	0.006	0.0046	0.0037	0.0031	0.0026	0.0023
0	$1 + \alpha_{TI}^{high}$	0.4375	0.3321	0.2705	0.2268	0.1929	0.1651
1	0.0914	0.04	0.0304	0.0247	0.0207	0.0176	0.0151
2	0.1757	0.0769	0.0584	0.0475	0.0398	0.0339	0.029
3	0.2386	0.1044	0.0792	0.0645	0.0541	0.046	0.0394
4	0.2129	0.0931	0.0707	0.0576	0.0483	0.0411	0.0352
5	1.0986	0.0431	0.0327	0.0267	0.0224	0.019	0.0163
Freq\Time	7	8	9	10	11	12	13
-1	0.0019	0.0017	0.0014	0.0012	0.001	0.0008	0.0007
0	0.1417	0.1214	0.1035	0.0875	0.073	0.0598	0.0476
1	0.013	0.0111	0.0095	0.008	0.0067	0.0055	0.0044
2	0.0249	0.0213	0.0182	0.0154	0.0128	0.0105	0.0084
3	0.0338	0.029	0.0247	0.0209	0.0174	0.0143	0.0114
4	0.0302	0.0258	0.022	0.0186	0.0155	0.0127	0.0101
5	0.014	0.012	0.0102	0.0086	0.0072	0.0059	0.0047
Freq\Time	14	15	16				
-1	0.0005	0.0004	0.0002				
0	0.0364	0.0259	0.0161				
1	0.0033	0.0024	0.0015				
2	0.0064	0.0045	0.0028				
3	0.0087	0.0062	0.0038				
4	0.0077	0.0055	0.0034				
5	0.0036	0.0026	0.0016				

efficients (MFCC). The MFCC code from the VoiceBox Toolkit [124] is used for algorithm implementation. The baseline results are based on the standard 13 MFCCs together with the corresponding velocity and acceleration parameters, denoted as MFCC(39). Figure 5.3 shows the diagram of the proposed algorithm.

As discussed in Chapter 4, the Adaptive 2D Psychoacoustic filter is the final form of the 2D psychoacoustic filter series. Therefore, only the experimental results of Adaptive 2D Psychoacoustic filter are provided in detail (Table 5.3 and Table 5.4). Table 5.6 gives the results of TW 2D filter and TFW 2D filter.

Figure 5.3: System Diagram of the Proposed Algorithm

Table 5.3: Recognition Results of Adaptive 2D L-Filter (%) for Clean Training Condition.

	Noise Type	Clean	20	15	10	5	0	-5	Avg 0-20
Set A	Subway	99.20	97.64	96.04	92.51	84.53	64.75	30.98	87.09
	Babble	99.18	98.16	96.98	94.07	81.89	54.53	22.79	85.13
	Car	99.28	98.30	96.39	92.54	82.88	57.80	22.73	85.58
	Exhibition	99.66	97.78	94.82	89.73	77.26	56.06	29.77	83.13
Set B	Restaurant	99.20	98.37	97.27	93.52	83.70	57.66	27.30	86.10
	Street	99.18	97.85	96.46	92.81	82.80	60.91	26.87	86.17
	Airport	99.28	98.45	97.46	94.09	83.15	57.83	25.02	86.20
	Train	99.66	98.21	96.48	92.90	83.49	56.68	23.17	85.55
Set C	Restaurant	99.29	97.18	95.89	91.80	80.99	56.98	24.50	84.57
	Street	99.12	97.73	96.19	91.38	79.84	56.26	22.52	84.28
Avg		99.31	97.97	96.4	92.54	82.05	57.95	25.57	85.38

5.4.2 Results Analysis

Table 5.5 shows the experimental results of MFCC(39), forward masking (FM), lateral inhibition (LI), and CMVN. From the experimental results, it can be easily seen that the proposed algorithm holds advantage over the comparison targets. As shown in the table, for all the above mentioned algorithms, the clean testing set results are over 99%. It is thus not very meaningful to discuss improvements at such level. Therefore, only the noisy test set results are used for comparison. The first set of comparison is made based on the clean training condition results. For MFCC(39), the relative improvements are 0.62% in SNR of 20 dB, 3.09% in SNR of 15 dB, 14.02% in SNR of 10 dB, 46.47% in SNR of 5 dB, 104.12% in SNR of 0 dB, and 96.09% in SNR of -5 dB. When compared with other algorithms, the proposed

Table 5.4: Recognition Results of Adaptive 2D L-Filter (%) for Multi Training Condition.

	Noise Type	Clean	20	15	10	5	0	-5	Avg 0-20
Set A	Subway	99.02	98.46	97.85	96.28	93.80	81.55	54.25	93.59
	Babble	98.79	98.82	98.28	97.1	92.32	75.24	42.14	92.35
	Car	98.78	98.54	98.06	96.69	92.34	78.59	44.02	92.84
	Exhibition	99.17	98.55	97.53	95.09	89.42	74.14	48.19	90.96
Set B	Restaurant	99.02	98.96	98.46	97.54	92.66	77.03	45.29	92.93
	Street	98.79	98.37	97.67	96.1	90.87	77.24	45.62	92.05
	Airport	98.78	98.48	98.24	97.11	93.11	79.78	47.81	93.34
	Train	99.17	98.95	98.12	96.85	92.47	76.61	44.89	92.60
Set C	Restaurant	99.05	98.13	97.64	95.95	92.11	78.14	46.58	92.39
	Street	99.00	98.43	97.94	95.98	90.05	74.82	40.72	91.44
Avg		98.96	98.57	97.98	96.47	91.92	77.31	45.95	92.45

algorithm yields better results. In terms of Avg 0-20, the relative improvements in terms of recognition rate are 19.76% over MFCC(39), 10.40% over FM, 15.43% over LI, and 9.77% over CMVN. For the SNR -5 condition, the relative improvements are much larger, 96.09% over MFCC(39), 20.05% over FM, 49.79% over LI, and 83.96% over CMVN.

Table 5.5: Recognition results for comparison targets (%).

	SNR/dB	Clean	20	15	10	5	0	-5	Avg 0-20
Clean Training	MFCC(39)	99.36	97.37	93.51	81.16	56.02	28.39	13.04	71.29
	FM	99.03	97.02	93.91	85.89	68.24	41.65	21.30	77.34
	LI	99.42	97.19	94.23	83.29	60.92	34.21	17.07	73.97
	CMVN	99.32	96.97	94.32	87.59	71.20	38.84	13.90	77.78
Multi Training	MFCC(39)	99.11	98.18	97.60	95.52	87.61	60.37	26.83	87.85
	FM	98.74	98.16	97.47	95.25	87.19	59.32	25.46	87.48
	LI	99.13	98.19	97.62	95.53	88.06	61.93	26.59	88.26
	CMVN	98.94	98.51	97.89	96.27	91.06	74.81	42.63	91.71

In the multi training condition, both clean and noisy speech are used for HMM training. Therefore, the recognition rate is very high compared with the clean training condition. However, the proposed algorithm still manages to achieve satisfying

improvements. Compared with MFCC(39), the relative improvements are 0.40% in SNR of 20 dB, 0.39% in SNR of 15 dB, 0.99% in SNR of 10 dB, 4.92% in SNR of 5 dB, 28.06% in SNR of 0 dB, and 71.26% in SNR of -5 dB. The relative improvements in terms of Avg 0-20 are 5.24% over MFCC(39), 5.68% over FM, 4.75% over LI, and 0.81% over CMVN. For the SNR -5 condition, the relative improvements are much better, 80.48% over FM, 72.81% over LI, and 7.79% over CMVN.

The second set of comparison is made against the corresponding H-filters. Table 5.6 gives the experimental results. It can be seen that the conversion from H-filter to L-filter helps the ASR system to achieve better or comparable results. For example, in terms of Avg 0-20 all the L-filter outperforms the corresponding H-filters.

Table 5.6: Comparison of H-filter and L-filter (Avg 0-20).

	SNR	TW 2D		TFW 2D		Adaptive 2D	
		clean	multi	clean	multi	clean	multi
H-filter	Avg 0-20	80.36	91.45	84.40	91.81	85.28	92.44
	-5 dB	14.41	38.57	24.37	43.86	24.67	46.13
L-filter	Avg 0-20	80.66	91.58	84.59	91.84	85.38	92.45
	-5 dB	14.66	37.49	24.06	43.29	25.57	45.95

5.5 Conclusion

In this chapter, a different way of implementing the psychoacoustic effect is proposed. Instead of removing the verbose audio signal, the signals (below the masking threshold) are reassigned to the optimal value (in terms of recognition rate) based on the total amount of masking. The new filters are all low-pass filters and are thus named 2D psychoacoustic L-filters. The proposed algorithm is developed based on the MFCC feature extraction algorithm. Since the L-filter can be developed from the corresponding H-filter, it possesses all the advantages of the H-filter introduced

in Chapter 4, such as asymmetry and frequency adaptation.

Chapter 6

Multiple Model Feature Compensation

6.1 Introduction

Noise robustness is the common problem of speech enhancement and speech recognition. Speech enhancement is a relatively well developed topic [4, 10, 13–15, 26, 96, 131, 132]. Therefore, the migration of noise removal algorithms from speech enhancement to speech recognition is a very common approach. Some try to use an additional black box approach, and some combine the speech enhancement algorithm within the feature extraction algorithm [13, 15, 26, 96]. Traditional speech enhancement algorithms, e.g. Spectral Subtraction, Wiener filtering, and Minimum Mean Square Error (MMSE), can be directly implemented as part of the feature extraction algorithm [36, 38–40, 133]. These algorithms nearly obtain the optimal results under their specific assumptions. The solid mathematical derivation makes it very easy for analysis and further development.

The difficulty of noise reduction in speech processing lies in that the statistical property of noise is unpredictable (or very difficult to predict). It is because there are numerous types of noise. The probability distribution function (PDF) of noise varies for different situations. When the background noise is the conversation of other people, the noise (background conversation) and the target speech share nearly the same statistical property. One straight forward way to solve the above mentioned problem is to obtain additional information of the noise. The stereo data based algorithms are good examples. Normally, noise removal algorithms need the training material to be processed by the same algorithm to make up for the distortion caused by the target algorithms. However, the stereo data based algorithms can work merely on the Hidden Markov Model (HMM) trained from the original clean speech, which means the algorithm indeed brings the noisy speech ‘closer’ to the clean speech [16, 45].

According to Equation (2.6), if $|Y(f, t)|^2$ and $|\tilde{N}(f, t)|^2$ are known. With no additional constraints, the clean speech can be estimated by

$$|\hat{X}(f, t)|^2 = |Y(f, t)|^2 - |\tilde{N}(f, t)|^2, \quad (6.1)$$

where $|\hat{X}(f, t)|^2$ is the clean speech power estimation.

Equation (6.1) is the basic idea of Spectral Subtraction (SS) [38]. For a typical speech signal, it is very rare to possess a completely quiet period (0 dB speech power region). Even for the silence period recorded in a clean room ($\text{SNR} \gg 20$ dB), there still would be very weak background noise due to equipment limitations. Therefore, speech power should be positive (in dB). Thus, the relationship in Equation (6.1) is only valid when $|Y(f, t)|^2 > |\tilde{N}(f, t)|^2$. If $|Y(f, t)|^2 \leq |\tilde{N}(f, t)|^2$, then $|Y(f, t)|^2 - |\tilde{N}(f, t)|^2 \leq 0$. The clean speech estimate becomes zero or negative, which is obviously wrong (in terms of physical meaning). A traditional way to solve

the above mentioned problem is to implement a threshold to guarantee the clean speech power estimate to be positive.

$$|\hat{X}(f, t)|^2 = \max \left[|Y(f, t)|^2 - |\tilde{N}(f, t)|^2, \varepsilon \right], \quad (6.2)$$

where the parameter ε is a small positive constant value.

Figure 6.1: Effect of Noise on Speech Power Spectrum.

Equation (6.2) is a very common way to implement Spectral Subtraction (SS) in speech recognition which will be denoted as SS in later discussion. Admittedly, Equation (6.2) manages to increase SNR, which should be a straight forward way to improve the performance of speech recognition systems. However, there is a very serious problem caused by SS. Due to the threshold, ε , a certain portion of the recovered speech is corrected to ε . Figure 6.2 gives an example of the effect of SS on speech spectrogram. The blue area in Figure 6.1(b) is all equal to ε .

After being processed by SS or other similar methods, the probability distribution of the speech is greatly changed. For example, in Figure 6.2, it can be easily seen that the probability of speech power which is equal to ε is greatly increased, which makes the PDF of the processed speech discontinuous. Most state-of-the-art ASR systems incorporate statistical methods to perform pattern recognition, e.g. HMM and ANN.

(a) Clean Speech PDF

(b) Noisy Speech PDF

(c) Speech After SS

Figure 6.2: Effect of SS on Speech Histogram.

These statistical methods are developed based on certain statistical assumption of the speech. In other words, a probability distribution is always assumed as the basis of recognizer derivation. The SS series algorithms greatly change the PDF of the speech, which causes the performance of ASR systems to drop. This chapter introduces two front-end noise reduction algorithms, Smoothing and Noise Subtraction (SNS) as well as Newton and Log Power Subtraction (NLPS). The proposed algorithms try to solve the discontinuity problem.

6.2 Smoothing & Noise Subtraction (SNS)

6.2.1 General Description

The histogram of speech log power spectrum shows that the contamination (or addition) of noise usually leads to a shift in the noise peak, which leads to the ambiguity between speech and noise. For example, the peak at about 18 dB in Figure 6.2(a) moves to about 48 dB in Figure 6.2(b). Therefore, a two-step feature enhancement scheme is proposed, which separates speech and noise peaks and yields positive results. Unlike the clean speech estimation approach, no complicated mathematical derivation is needed. By analyzing speech from a different angle, a simple but effective method is developed which improves the performance of ASR systems.

Figure 6.3 shows the distribution of the speech power spectrum along frames, and Figure 6.4 gives an example of speech PDF for clean and noisy speech samples. Figure 6.3 gives an idea of how the average speech power within a frame changes as the amount of noise increases. A better perspective can be seen from the histogram given in Figure 6.4. It can be seen that speech PDF consists of two parts, pure speech and noise. It has to be noted that even for clean speech, there is always

(a) Clean Speech

(b) Noisy Speech

Figure 6.3: Effect of Noise on Speech Power Spectrum.

some noise in the speech. The so called clean speech is just noisy speech with very high SNR.

(a) Clean Speech

(b) Noisy Speech

Figure 6.4: Effect of Noise on Speech PDF.

It can be seen that the increase of noise power mainly causes the noise peak to shift closer to the speech peak. The noise peak goes from about 18 dB in Figure 6.4(a) to 48 dB in Figure 6.4(b). The shifting of noise peak causes the ambiguity between noise and speech. For example, the difference between speech and noise is relatively clear in Figure 6.4(a). Generally, speech component with power larger than 50 dB can be treated as speech, which means with proper settings most of

the speech can be clearly identified by a statistical recognizer. However, when it comes to noisy speech the difference between noise and speech becomes vague. For example, it is very hard to decide whether a noisy speech component of 70 dB is noise or speech from Figure 6.4(b). This significantly degrades the performance of automatic speech recognition systems.

Figure 6.5: Overlapping of Two PDFs.

Figure 6.5 gives another example of the above mentioned phenomena by using two Gaussian curves. When the two distributions are relatively far from each other, there is little overlap between them which is the case for clean speech. When the two distributions become closer to each other, it can result in a large amount of overlap, which reflects the case of noisy speech. For Figure 6.5(a), there is little difficulty for a statistical recognizer to tell which distribution a given value belongs to. However, when it comes to Figure 6.5(b) the huge overlapping area makes the problem very difficult.

The proposed algorithm intends to solve the above mentioned overlapping problem by a two-step scheme, i.e. smoothing and noise subtraction (SNS). The two-step scheme is developed to recover the relative position of speech and noise peaks so

Figure 6.6: Block diagram of SNS.

as to reduce the amount of overlapping between the two. Figure 6.7 gives a brief explanation of the above mentioned process using two Gaussian curves. The original probability distribution function (PDF) is the combination of two statistical processes, shown as two Gaussian curves D1 and D2. The combination curve is denoted as D3. Owing to the overlapping region, it is very hard to tell which model the value that falls in the grey region belongs to. One straight forward idea is to move noise peak and speech peak away from each other. The idea can be realized by first reducing the variance using the Moving Average (MA) filter and then going through the Noise Subtraction (NS) to remove the noise (shift the noise peak away from the speech peak).

6.2.2 2D Smoothing

From Equation (2.6), the relationship between the noisy speech power, noise power and pure speech power can be modeled as

$$|Y(f, t)|^2 = |X(f, t)|^2 + |\tilde{N}(f, t)|^2. \quad (6.3)$$

In order to solve the overlapping problem, one straight forward way is to reduce the variance of both distributions and shift the noise peak towards the left side (weaken the noise), which correspond to the two parts of the proposed algorithm

Figure 6.7: Example of the proposed algorithm.

(smoothing and noise subtraction). This section discusses the smoothing process. It is well known that MA filter can effectively reduce the variance of random variables. Therefore, MA filter is adopted in our proposed algorithm. In statistics, a moving average filter is a type of finite impulse response filter, which averages a number of input samples and produce a single output sample. In this paper a 2D MA filter is implemented as

$$\tilde{Y}(f, t) = \frac{\sum_{j=-d}^d \sum_{i=-d}^d Y(f+i, t+j)}{(2d+1)^2}. \quad (6.4)$$

Let $M(\cdot)$ denotes the MA filtering process, then

$$\begin{aligned} & M[|Y(f, t)|^2] \\ &= M[|X(f, t)|^2 + |\tilde{N}(f, t)|^2] \\ &= M[|X(f, t)|^2] + M[|\tilde{N}(f, t)|^2]. \end{aligned} \quad (6.5)$$

Speech and noise power spectra follow certain probability distribution with parameter (μ_x, σ_x^2) and (μ_n, σ_n^2) respectively. After MA filtering, pure speech variance σ_x^2 and noise variance σ_n^2 become $1/(2d+1)^2$. In our present implementation, d

is chosen to be 1, which is obtained empirically. After processed by a 3×3 MA filter, the PDF becomes more concentrated around the mean since σ_x^2 and σ_n^2 have reduced to 1/9 of the original values.

6.2.3 Noise Subtraction

A common way to solve the negative value problem discussed in Section 6.1 is by the adoption of a threshold

$$\left| \hat{X}(f, t) \right|^2 = \max(|Y(f, t)|^2 - \alpha|N(f, t)|^2, \varepsilon), \quad (6.6)$$

where ε is a small constant number.

Equation (6.6) is a commonly used way to implement Spectral Subtraction (SS) in speech recognition and will be denoted as ε -SS in later discussion. Admittedly, Equation (6.6) manages to remove a large amount of noise and increase the signal to noise ratio (SNR). However, the problem caused by ε -SS is also very serious. Because of the threshold, ε , a certain portion of the speech (negative part) is set to be ε . The amount of negative speech components is dependent on the SNR of the noisy speech. For high SNR speech (20 dB or larger) the number can be 10% or less of the overall speech. However, for low SNR speech (-5 dB or smaller), the number can be very high, about 20% or larger. Figure 6.2 gives an example of the effect of ε -SS on speech spectrogram. The blue area in Figure 6.1(b) consists of large regions equal to ε . Although noise energy is suppressed after being processed by ε -SS, speech is also distorted from the original form. The probability distribution of speech is greatly changed. As shown in Figure 6.8, the probability of speech power equal ε is greatly increased, which causes the discontinuity problem.

Statistical recognizers, such as HMM and ANN, are very commonly used in state-of-the-art automatic speech recognition systems. Certain probability distribution

Figure 6.8: Speech PDF: digit string ‘3Z82’ from the AURORA2 database.

functions are always assumed for the speech and noise as the basis for algorithm derivation. However, ε -SS introduces serious interruption to the system, which causes the performance of ASR systems to drop.

Actually, the problem comes mainly from the massive assignment of the negative speech components to a constant number, i.e. ε . The discontinuity problem can be solved by replacing the negative numbers with random number generated by certain statistical model. The idea comes from the observation that even the clean speech contains very weak noise, which appears to have little impact on the final results.

Table 6.1: Recognition results for comparison targets under multi training condition (%)

	Clean	Avg 0-20	-5
CMVN	99.32	77.78	13.90
ε -SS	95.11	72.42	25.26
Gaussian-SS	98.89	79.66	25.43
o-SS	98.98	80.13	24.72
a-SS	98.48	86.95	32.56

A better way to deal with the negative values is to assign them to the noise

area (around the noise peak). There are many different ways to achieve the above mentioned goal. Table 6.1 shows the results of different negative value compensation methods. In the table, ε -SS stands for the algorithm that changes all negative numbers to a constant ε . Gaussian-SS denotes the one that use Gaussian random number to replace the negative values. o-SS stands for the approach that utilizes the original speech component (before SS) to replace the negative value. Finally, a-SS stands for the adoption of absolute value of the negative values. Cepstral Mean and Variance Normalization (CMVN) is implemented in all systems.

It can be easily seen that using small constant number actually makes the speech recognition result worse. For the random number approach, the result shows some improvement, 79.66%, which is better than CMVN. The improvement indicates that a suitable statistical compensation is better than the constant number approach. The next approach, o-SS, attempts to derive the compensation number directly from the noisy speech. As expected, further improvement is achieved from o-SS. The success of o-SS indicates that derivation of compensation numbers from the speech itself is a promising approach.

Although o-SS leads to some improvements, the problem lies in that the replacement of negative value using the original speech is a waste of the SS noise removing scheme. When it comes to a-SS, it adopts the absolute value of the negative components as the compensation estimate. The negative components share similar properties as the noise region. Therefore, a-SS can satisfactorily reassign the negative components to the noise region, which helps to improve the performance of speech recognition system.

6.3 Newton & Log Power Subtraction (NLPS)

6.3.1 General Description

Based on the MFCC feature extraction algorithm discussed in Section 2.2.1, we define the log channel energy vectors as,

$$\mathbf{Y} = \begin{bmatrix} \log \sum_f W_f^1 |Y(f, t)|^2 \\ \log \sum_f W_f^2 |Y(f, t)|^2 \\ \vdots \\ \log \sum_f W_f^L |Y(f, t)|^2 \end{bmatrix} \quad (6.7)$$

$$\mathbf{X} = \begin{bmatrix} \log \sum_f W_f^1 |X(f, t)|^2 \\ \log \sum_f W_f^2 |X(f, t)|^2 \\ \vdots \\ \log \sum_f W_f^L |X(f, t)|^2 \end{bmatrix} \quad (6.8)$$

$$\mathbf{N} = \begin{bmatrix} \log \sum_f W_f^1 |N(f, t)|^2 \\ \log \sum_f W_f^2 |N(f, t)|^2 \\ \vdots \\ \log \sum_f W_f^L |N(f, t)|^2 \end{bmatrix}, \quad (6.9)$$

where $\log(\cdot)$ denotes the natural logarithm as defined in Chapter 2.

Then Equation (2.7) becomes

$$e^{\mathbf{Y}} = e^{\mathbf{X}} + e^{\mathbf{N}}. \quad (6.10)$$

Changing Equation (6.10) to the log-power domain

$$\mathbf{Y} = \log(e^{\mathbf{X}} + e^{\mathbf{N}}). \quad (6.11)$$

Then

$$\mathbf{Y} = \mathbf{X} + \log (\mathbf{1} + e^{\mathbf{N}-\mathbf{X}}), \quad (6.12)$$

where $\mathbf{1}$ stands for a vector with all components equal to one.

6.3.2 Iterative Solution

The novelty of implementing iterative root finding algorithm is that unlike the SS series approaches, it manages to overcome the awkward negative value problem without causing discontinuity in the speech PDF. The statistical approach handles this by applying a series of mathematical operations which are not sensitive to the above mentioned problem. In power domain, the final expression is

$$|\hat{X}(f, t)|^2 = G \times |Y(f, t)|^2 = |Y(f, t)|^2 - (1 - G) \times |Y(f, t)|^2, \quad (6.13)$$

which fundamentally avoids the possibility of $|Y(f, t)|^2 \leq |N(f, t)|^2$. It is because the equivalent noise estimate is $(1 - G) \times |Y(f, t)|^2$, which is generated from only the current frame.

As described before, the iterative root finding algorithm can also handle $|Y(f, t)|^2 \leq |N(f, t)|^2$ very well. Equation (6.12) can be reshaped to

$$\mathbf{X} + \log (\mathbf{1} + e^{\mathbf{N}-\mathbf{X}}) - \mathbf{Y} = 0, \quad (6.14)$$

where $\mathbf{Y}$ is the noisy speech vector, $\mathbf{X}$ is the parameter that is needed to be recovered. If the noise vector, $\mathbf{N}$, can be reasonably estimated, Equation (6.14) becomes a nonlinear function about $\mathbf{X}$, which can be solved by iterative root finding algorithms.

Denoting

$$f(\mathbf{X}) = \mathbf{X} + \log (\mathbf{1} + e^{\mathbf{N}-\mathbf{X}}) - \mathbf{Y}. \quad (6.15)$$

Then

$$f'(\mathbf{X}) = \frac{d[f(\mathbf{X})]}{d\mathbf{X}} = \mathbf{1} - \frac{e^{\mathbf{N}-\mathbf{X}}}{1 + e^{\mathbf{N}-\mathbf{X}}}. \quad (6.16)$$

According to Newton's method, given a function $f(\mathbf{X})$, its derivative $f'(\mathbf{X})$ and a first guess $\hat{\mathbf{X}}_0$, the solution to the function can be calculated by

$$\mathbf{X}_{j+1} = \mathbf{X}_j - \frac{f(\mathbf{X}_j)}{f'(\mathbf{X}_j)}, \quad (6.17)$$

where j is the iteration index.

For the iterative step

$$\mathbf{X}_{j+1} = \mathbf{X}_j - \frac{\mathbf{X}_j + \log(\mathbf{1} + e^{\mathbf{N}-\mathbf{X}_j}) - \mathbf{Y}}{\mathbf{1} - \frac{e^{\mathbf{N}-\mathbf{X}_j}}{1+e^{\mathbf{N}-\mathbf{X}_j}}}. \quad (6.18)$$

It has to be noted that

$$\lim_{(\mathbf{N}-\mathbf{X}_i) \rightarrow +\infty} \mathbf{1} - \frac{e^{\mathbf{N}-\mathbf{X}_i}}{\mathbf{1} + e^{\mathbf{N}-\mathbf{X}_i}} = 0. \quad (6.19)$$

Therefore, a threshold, β , is adopted to guarantee the denominator to be non-zero. Then Equation (6.18) is modified to

$$\mathbf{X}_{i+1} = \mathbf{X}_i - \frac{\mathbf{X}_i + \log(\mathbf{1} + e^{\mathbf{N}-\mathbf{X}_i}) - \mathbf{Y}}{\max\left(\mathbf{1} - \frac{e^{\mathbf{N}-\mathbf{X}_i}}{1+e^{\mathbf{N}-\mathbf{X}_i}}, \beta\right)}. \quad (6.20)$$

With a successful guess of the initial step, clean speech vector and noise vector, the clean speech estimate can be satisfactorily approximated. Equation (6.20) can work very well even if $|Y(f, t)|^2 < |\tilde{N}(f, t)|^2$. About the discontinuity problem, at extreme conditions where the threshold β works, the iteration becomes

$$\begin{aligned} & \mathbf{X}_{i+1} \\ = & \mathbf{X}_i - \frac{\mathbf{X}_i + \log(\mathbf{1} + e^{\mathbf{N}-\mathbf{X}_i}) - \mathbf{Y}}{\beta} \\ = & (1 - \beta) \mathbf{X}_i - \frac{1}{\beta} \log(\mathbf{1} + e^{\mathbf{N}-\mathbf{X}_i}) + \frac{1}{\beta} \mathbf{Y}. \end{aligned} \quad (6.21)$$

It can be easily seen that Equation (6.21) will not cause mass assignment of the same value, which means the discontinuity problem will not appear.

6.3.3 Prior Estimates

In statistics, a minimum mean square error (MMSE) estimator is the approach to minimize the mean square error (MSE), which is widely used in many areas in signal processing. In 1984, Ephraim derived the short time spectral amplitude (STSA) estimator using MMSE [36]. After that, MMSE has become a standard approach for enhancing the quality of speech. Therefore, it is chosen to generate the prior estimate of the clean speech. The following equation shows the standard cost function for MMSE approach [36]

$$\hat{X} = \arg \min_{\hat{X}} E \left[\left(X - \hat{X} \right)^2 \right]. \quad (6.22)$$

By following MMSE-STSA [36] the clean speech estimate can be calculated by

$$\begin{aligned} & \hat{X}(f, t) \\ = & \Gamma(1.5) \frac{\sqrt{v(f, t)}}{\gamma(f, t)} M[-0.5; 1; -v(f, t)] Y(f, t) \\ = & \Gamma(1.5) \frac{\sqrt{v(f, t)}}{\gamma(f, t)} \exp \left[-\frac{v(f, t)}{2} \right] \times \\ & \left\{ [1 + v(f, t)] I_0 \left[\frac{v(f, t)}{2} \right] + v(f, t) I_1 \left[\frac{v(f, t)}{2} \right] \right\} Y(f, t), \end{aligned} \quad (6.23)$$

where $\Gamma(\cdot)$ denotes the gamma function; $M(a; c; x)$ is the confluent hypergeometric function; $I_0(\cdot)$ and $I_1(\cdot)$ denote the zero and first order modified Bessel function; $\xi(f, t)$ and $\gamma(f, t)$ are the a-priori and a-posteriori signal-to-noise ratios (SNR), respectively.

$$v_{f,t} = \frac{\xi_{f,t}}{1 + \xi_{f,t}} \gamma_{f,t}. \quad (6.24)$$

Then the clean amplitude estimate is transferred to log-power domain,

$$\hat{X}^l = \log \sum_f W_f^l \left| \hat{X}(f, t) \right|^2. \quad (6.25)$$

Equation (6.25) will serve as the initial guess for the iteration approach.

Figure 6.9: Diagram of NLPS.

6.3.4 Log-Power Subtraction (LPS)

The proposed algorithm works in the log-power domain. The clean speech estimate generated by the proposed algorithm in Equation (6.20) is actually

$$\mathbf{X} = \begin{bmatrix} \log \sum_f W_f^1 |X(f, t)|^2 \\ \log \sum_f W_f^2 |X(f, t)|^2 \\ \vdots \\ \log \sum_f W_f^L |X(f, t)|^2 \end{bmatrix}. \quad (6.26)$$

It is the clean speech log-power vector in the log-power domain. The MFCC static parameters can be divided into two parts, $c1 \sim c12$ and $c0/\log\text{-energy}$. Strictly

speaking, the proposed algorithm mainly focuses on c1~c12. For log-energy, traditionally it should be calculated by

$$P_{\log} = \log \left[\sum_f |X(f, t)|^2 \right]. \quad (6.27)$$

The clean speech power estimate, $|X(f, t)|^2$, cannot be perfectly recovered from Equation (6.26) because of the Mel-filterbanks. Additional distortion will be introduced to the feature vectors. For C_0 , although it seems to work smoothly, the recognition results are just about ‘average’. Therefore, a separate noise removing scheme is developed. At the iterative root finding part, an estimate for the noise is reached.

Then the log-energy can be calculated by

$$\hat{P}_c = \sum_f \left[|Y(f, t)|^2 - \hat{N}(f, t) \right]. \quad (6.28)$$

Then the log-energy can be calculated by

$$P_{\log} = \log \left(\hat{P}_c \right) = \log \left[\sum_f \left(|Y(f, t)|^2 - \hat{N}(f, t) \right) \right]. \quad (6.29)$$

However, problem arises when $|Y(f, t)|^2 < \hat{N}(f, t)$. Therefore, a weighting parameter is incorporated to reduce the chances of imaginary parts appearing. Then Equation (6.29) becomes

$$\hat{P}_c = \sum_f \left[|Y(f, t)|^2 - \alpha \hat{N}(f, t) \right]. \quad (6.30)$$

Furthermore, another parameter ε_0 is set to ensure the log-energy will not be infinity. Therefore, the log-power part becomes

$$P_{\log} = \log \left(\hat{P}_c \right) = \log \left\{ \max \left[\sum_f \left(|Y(f, t)|^2 - \alpha \hat{N}(f, t) \right), \varepsilon_0 \right] \right\}. \quad (6.31)$$

6.4 Results and Discussion

6.4.1 Experimental Results

Evaluation tests are carried out using the AURORA2 database. Detailed experimental results are given in Tables 6.2, 6.3, 6.6, and 6.7. Experimental results are averaged over SNR of 0 dB to 20 dB denoted as Avg 0-20. ‘Rel. Imp.’ stands for relative improvements in terms of recognition rate.

Table 6.2: Recognition Results of SNS (%) for Clean Training Condition

	Noise Type	Clean	20	15	10	5	0	-5	Avg 0-20
Set A	Subway	98.34	97.94	97.05	94.17	88.15	73.32	44.43	90.13
	Babble	98.70	98.04	97.10	94.95	87.45	67.23	34.22	88.95
	Car	98.63	97.97	97.41	95.17	89.26	71.79	41.16	90.32
	Exhibition	98.55	97.87	96.33	91.55	82.04	63.53	39.31	86.26
Set B	Station	98.34	97.94	97.24	94.50	86.80	66.41	35.16	88.58
	Restaurant	98.70	97.55	96.80	93.20	85.31	66.44	37.12	87.86
	Street	98.63	98.09	97.41	95.53	88.67	70.15	36.77	89.97
	Airport	98.55	98.24	96.7	95.22	89.08	70.90	40.79	90.03
Set C	Restaurant	98.25	97.64	96.75	93.43	85.35	65.55	34.17	87.74
	Street	98.49	97.85	96.28	92.35	81.98	61.94	33.62	86.08
Avg		98.52	97.91	96.91	94.01	86.41	67.73	37.69	88.59

Table 6.3: Recognition Results of SNS (%) for Multi Training Condition

	Noise Type	Clean	20	15	10	5	0	-5	Avg 0-20
Set A	Subway	97.94	98.00	97.76	96.28	93.21	82.07	55.73	93.46
	Babble	98.31	98.28	97.58	97.28	92.41	75.91	45.13	92.29
	Car	98.09	98.69	98.18	97.14	94.18	83.66	56.58	94.37
	Exhibition	98.40	98.33	97.56	95.37	89.36	75.32	51.16	91.19
Set B	Station	97.94	98.34	97.82	96.78	91.25	74.21	43.81	91.68
	Restaurant	98.31	98.07	97.79	95.74	91.11	77.63	48.55	92.07
	Street	98.09	98.18	98.03	96.54	93.05	80.02	50.37	93.16
	Airport	98.4	98.55	97.87	97.38	92.63	80.81	54.86	93.45
Set C	Restaurant	97.70	97.76	97.36	95.67	91.37	77.96	49.09	92.02
	Street	97.79	98.07	97.67	95.50	90.36	75.21	45.13	91.36
Avg		98.10	98.23	97.76	96.37	91.89	78.28	50.04	92.51

For SNS, the algorithm consists of three parts, noise subtraction, 2D smooth-

ing, and CMVN. Table 6.4 gives the clean training condition results for NS, 2D smoothing, and CMVN. It can be seen that all parts contribute to the final results.

Table 6.4: Experimental Results for Different Parts of SNS

	Clean	Avg 0-20	-5
CMVN	99.32	77.78	13.90
NS	98.47	86.95	32.56
2D Smoothing	99.29	81.45	22.69
SNS	98.53	88.59	37.69

For NLPS, experiments are conducted to show the speech recognition results of the proposed NLPS algorithm with different iteration numbers. Since the multi training condition results are all around 92%, only the clean training condition results are used for optimization. Experimental results are given in Table 6.5. It can be easily seen that the optimal result can be obtained at the second iteration.

Table 6.5: Experimental Results for Different Iterations

Iteration No.	1	2	3	4	5
clean	99.09	99.08	98.37	98.36	97.23
Avg0-20	85.70	86.11	85.66	80.07	76.17
-5 dB	27.80	28.57	23.24	23.24	21.31

Table 6.6: Recognition Results of NLPS (%) for Clean Training Condition

	Noise Type	Clean	20	15	10	5	0	-5	Avg 0-20
Set A	Subway	98.83	97.27	95.61	91.68	82.56	62.70	29.23	85.96
	Babble	99.09	98.22	97.04	94.35	84.31	59.70	26.18	86.72
	Car	99.14	98.21	96.75	93.71	85.09	65.08	31.40	87.77
	Exhibition	99.29	97.5	95.03	88.77	76.64	56.12	29.37	82.81
Set B	Restaurant	98.83	98.28	97.14	93.89	83.79	60.91	29.63	86.80
	Street	99.09	97.70	96.43	92.9	83.65	63.51	29.50	86.84
	Airport	99.14	98.48	97.38	95.02	85.80	64.96	31.26	88.33
	Train	99.29	98.43	96.85	94.45	86.55	63.44	30.64	87.94
Set C	Restaurant	98.93	97.42	95.76	91.03	79.09	52.44	21.80	83.15
	Street	99.15	97.52	96.01	91.96	82.01	56.59	26.72	84.82
Avg		99.08	97.90	96.40	92.78	82.95	60.55	28.57	86.11

Table 6.7: Recognition Results of NLPS (%) for Multi Training Condition

	Noise Type	Clean	20	15	10	5	0	-5	Avg 0-20
Set A	Subway	98.46	98.13	97.51	95.86	93.09	80.81	48.66	93.08
	Babble	98.31	98.19	97.94	97.04	91.78	73.64	36.37	91.72
	Car	98.24	98.36	98.00	96.63	93.14	82.40	47.09	93.71
	Exhibition	98.61	98.30	97.22	95.00	89.63	74.54	46.78	90.94
Set B	Restaurant	98.46	98.40	98.04	96.56	91.71	74.95	39.7	91.93
	Street	98.31	98.22	97.76	95.80	91.26	75.85	42.96	91.78
	Airport	98.24	98.57	98.12	96.69	92.90	79.09	44.26	93.07
	Train	98.61	98.64	97.78	96.88	92.50	78.83	47.86	92.93
Set C	Restaurant	98.37	98.00	97.30	95.36	91.22	74.09	38.32	91.19
	Street	98.19	97.79	97.64	95.83	90.02	71.70	36.88	90.60
Avg		98.38	98.26	97.73	96.17	91.73	76.59	42.89	92.09

Comparisons are made against MFCC(39), short-time spectral amplitude (STSA) based MMSE [36], ETSI standard advanced front-end feature extraction algorithm (AFE) [134], and MVA [47]. All the algorithms discussed in this thesis are developed based on MFCC. Therefore, it is chosen to be the baseline system. In this thesis, the MMSE algorithm is implemented with minimum statistics for noise estimation. Experimental results are given in Table 6.8.

Table 6.8: Recognition Results for Comparison Targets (%)

	SNR/dB	Clean	Avg 0-20	-5
Clean Training	MFCC(39)	99.36	71.29	13.04
	CMVN	99.32	77.78	13.90
	MMSE	99.26	80.12	20.31
	AFE	99.20	82.23	24.77
	MVA	99.20	84.15	26.24
Multi Training	MFCC(39)	99.11	87.85	26.83
	CMVN	98.94	91.71	42.63
	MMSE	98.74	92.25	48.61
	AFE	99.14	91.83	42.32
	MVA	98.94	92.22	49.23

6.4.2 Results Analysis

In this chapter, two different algorithms, NLPS and SNS, are proposed for noise removal. It can be seen that the proposed algorithms successfully improve the performance of automatic speech recognition systems. Since the clean test set results are all about 99%, it is meaningless to discuss improvements at such level. Therefore, the clean test set results are not used for comparison. For the clean training condition results, the relative improvement ratios are shown in Table 6.9. They are calculated in terms of recognition rate. It can be seen that the proposed algorithm performs better under both Avg 0-20 and SNR -5 dB conditions.

Table 6.9: Relative Improvements for Clean Training Condition (%)

SNR/dB	Avg 0-20		-5	
	SNS	NLPS	SNS	NLPS
MFCC(39)	24.27	20.79	189.03	119.10
CMVN	13.90	10.71	171.15	105.54
MMSE	10.57	7.48	85.57	40.67
AFE	7.73	4.72	52.16	15.34
MVA	5.28	2.33	43.64	8.88

At Avg 0-20, the relative improvements for NLPS are 20.79% over MFCC, 10.71% over CMVN, 7.48% over MMSE, 4.72% over AFE, and 2.33% over MVA. SNS performs a little bit better than NLPS. The relative improvements are 24.27% over MFCC, 13.90% over CMVN, 10.57% over MMSE, 7.73% over AFE, and 5.28% over MVA. Figure 6.10(a) gives a better view of the above mentioned results. When it comes to SNR of -5 dB, better results are obtained. The relative improvements of SNS are 189.03% over MFCC, 171.15% over CMVN, 85.57% over MMSE, 52.16% over AFE, and 43.64% over MVA. For NLPS, the relative improvements are 119.10% over MFCC, 105.54% over CMVN, 40.67% over MMSE, 15.34% over AFE, and

8.88% over MVA. The relative improvements are also given in Figure 6.10(b). SNS outperforms NLPS at all SNR levels.

Figure 6.10: Relative Improvements for Clean Training Condition.

Table 6.10: Relative Improvements for Multi Training Condition (%)

SNR/dB	Avg 0-20		-5	
	SNS	NLPS	SNS	NLPS
MFCC(39)	5.30	4.83	86.51	59.86
CMVN	0.87	0.41	17.38	0.61
MMSE	0.28	-0.17	2.94	-11.77
AFE	0.74	0.28	18.24	1.35
MVA	0.31	-0.14	1.65	-12.88

Since in the multi training condition both clean speech and noisy speech are used for HMM training, all the algorithms can yield very good recognition results. However, the proposed algorithms still achieve very positive results. For SNS, it performs better than all the other algorithms. At Avg 0-20, the relative improvements are 5.3% over MFCC(39), 0.87% over CMVN, 0.28% over MMSE, 0.74% over AFE, and 0.31% over MVA. At SNR -5 dB, better results are obtained, 86.51% over MFCC(39), 17.38% over CMVN, 2.94% over MMSE, 18.24% over AFE, and 1.65%

over MVA. For NLPS, the situation is a little bit different. It is better than MFCC, CMVN and AFE but slightly worse than MMSE and MVA.

6.5 Conclusion

In this chapter, two algorithms are proposed. SNS works by recovering the temporal structure of speech power spectrum. SNS analyzes the speech signal in a special way by the histogram of average speech log power spectrum within a frame. The proposed algorithm does not attempt to estimate the clean speech from noisy speech. It works on the histogram instead. A two-step scheme is proposed to weaken the noise effects by first reducing the noise variance and then shifting the noise mean. NLPS is based on the direct solution of a nonlinear system model together with a novel log-power subtraction method.

The proposed algorithms are blind approaches, which means that the proposed methods yield good performance at all SNRs and noise types. This property improves its ability to adapt to changing environments. Finally, both SNS and NLPS can be easily combined with any other algorithms. For example, MVA can be combined with the proposed algorithm by simply implementing an ARMA filter after CMVN. The effectiveness of the proposed algorithms is extensively tested using the AURORA2 database.

Chapter 7

Conclusion and Future Works

7.1 Conclusion

The key objective of robust speech recognition technology is to get better recognition accuracy, especially in noisy conditions. After years of development, state-of-the-art automatic speech recognition (ASR) systems can work very well with clean speech. For example, in Table 6.8, the recognition accuracies can be over 99% for the clean test set. However, with noise added in, the performance of ASR systems falls dramatically. One straight forward approach to solve the above mentioned problem is to remove noise before performing statistical pattern recognition.

In Chapter 2, a brief review of the general structure of automatic speech recognition systems is provided. Several front-end noise suppression algorithms are reviewed. Chapters 3, 4 and 5 introduce three different algorithms based on psychoacoustic models. The human auditory system can handle adverse situations very well. For example, in our daily life there are various kinds of noises and distortions such as environmental noise (conversation in a noisy street), channel distortion (telephone,

radio), and speaker variability (speak with different people). Normal people with a healthy auditory system have barely any difficulty in handling situations above. Therefore, the idea of analyzing and modeling the human auditory system is a logical approach to improve the performance of automatic speech recognition (ASR) systems.

Chapter 3 introduces the LTFC algorithm, which sequentially implement forward masking, simultaneous masking, and CMVN to the Mel-Frequency Cepstral Coefficients (MFCC) feature extraction algorithm. Mathematical derivations are provided to show the effectiveness of the proposed algorithm. Chapter 4 proposes a novel 2D psychoacoustic filter, which implements forward masking, lateral inhibition, and temporal integration at the same time. Theoretical analysis is provided to derive the 2D psychoacoustic filter from the characteristic functions of masking effects. The proposed method sharpens the spectrum of the signal in the time-frequency domain. Masking effects describe how a clearly audible sound become weak or inaudible in the presence of another sound. Therefore, all the sound components below the masking threshold are equivalent for the human auditory system, since they will be recognized as silence or nonexistence. The algorithm discussed in Chapter 4 tries to implement psychoacoustic models in a subtractive manner (the noisy speech minuses the estimated amount of masking), so as to remove noise. In Chapter 5, a different way of implementing psychoacoustic models is provided. It is implemented in an additive manner.

Another important part of the thesis is about feature compensation. Two different algorithms are proposed, i.e. Smoothing & Noise Subtraction (SNS) as well as Newton & Log Power Subtraction (NLPS). The histogram of average speech log power spectrum shows that the contamination of noise leads to a shift of the noise peak, which degrades the performance of speech recognition systems. The SNS al-

gorithm solves the problem by a two-step scheme, first reducing the noise variance and then shifting the noise mean. For NLPS, it is developed based on numerical root finding algorithms. It works by direct solution of the nonlinear function derived from the MFCC feature extraction algorithm. In contrast to the statistical clean speech estimation methods, the proposed algorithms do not need prior speech/noise statistical model, which makes it simple but effective.

7.2 Further Works

For the 2D psychoacoustic filter series, parameters of the diagonal region (temporal frequency masking) are derived from FM and LI parameters. It means that they may be different from the real parameters that the human auditory system adopts. The 2D psychoacoustic filters are proposed to model the human auditory system. Therefore, psychoacoustic experiments can be performed to obtain more accurate parameters for the 2D psychoacoustic filters.

Most state-of-the-art speech recognition systems use DFT to transform time domain speech signals to frequency domain. However, it has to be noted that in other speech related areas, researchers have experimented with other transformations, such as Discrete Cosine Transform (DCT). DCT has been proven to be a suitable alternative to DFT in speech enhancement [82, 135–137]. Therefore, it is logical to investigate the possibility of using DCT in speech recognition. Another example of time-frequency representation (TFR) is quadratic TFRs (QTFRs). It can be formulated by the multiplicative comparison of a signal with itself, expanded in different directions about each point in time. A detailed discussion on QTFRs with their property and application in speech recognition can be found in [138].

With limited information, it is very difficult to perfectly recover the clean speech

or effectively remove noise from the noisy speech. State-of-the-art noise robust speech recognition algorithms have already explored many different possible approaches to solve the problem and obtain very promising results. In order to push closer to the perfect solution, it is logical to incorporate more information to the system. The stereo data based algorithms are very good examples. Another way to gather more information is to use microphone array. Thus the input signal becomes different versions of the same signal, which makes the noise suppression problem much easier. With the sound signal captured from several points, beamforming or spatial filtering technique can be used for noise removal. Microphone array processing generally consists of two parts, sound source localizer and beamformer. The system firstly localize the sound source. Then, the system will amplify the desired signal coming from a specific direction, and attenuate signals from other directions, leading to a better SNR level [139].

Author's Publications

Journal Papers

1. P. Dai and I. Y. Soon, A Temporal Warped 2D Psychoacoustic Modeling for Robust Speech Recognition System. *Speech Communication* , Vol. 53, pp. 229–241, 2010.
2. P. Dai and I. Y. Soon, A Temporal Frequency Warped (TFW) 2D Psychoacoustic Filter for Robust Speech Recognition System. *Speech Communication*, Vol. 54, pp. 402–413, 2011.
3. P. Dai, I. Y. Soon, and R. Tao, Direct Recovery of Clean Speech using a Hybrid Noise Suppression Algorithm for Robust Speech Recognition System, *ISRN Signal Processing*, 2012.
4. P. Dai and I. Y. Soon, An Improved Model of Masking Effects for Robust Speech Recognition System. *Speech Communication*, vol 55, pp. 387-396, 2013.
5. P. Dai and I. Y. Soon, Robust Speech Recognition by Using Spectral Subtraction with Noise Peak Shifting. *IET Signal Processing*, vol. 7, pp. 684-692, 2013.

Conference Papers

1. P. Dai and I. Y. Soon and C. K. Yeo, 2D psychoacoustic filtering for robust speech recognition. in *Proc. International Conference on Information, Communications and Signal Processing (ICICS)*, pp. 1–5, 2009.
2. P. Dai and I. Y. Soon, Iterative Language Model Adaptation for Large Vocabulary Continuous Speech Recognition. in *Proc. IEEE International Conference on Computer Science and Information Technology (ICCSIT)* , Vol. 5, 2011.
3. P. Dai and I. Y. Soon, An Adaptive Soft Voice Activity Detector for Automatic Speech Recognition System. in *Proc. Internal Conference on Information, Communications and Signal Processing (ICICS)*, 2011.
4. B. Lei and I. Y. Soon and Z. Li and P. Dai, An Optimal Audio Watermarking Scheme Based on Particle Swarm Optimization. in *Proc. Internal Conference on Information, Communications and Signal Processing (ICICS)*, 2011.

5. P. Dai, W. Yang and S. S. Ho, "Predicting Mobile Call Behavior via Subspace Methods", in *Proc. International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction (SBP2013)*, Washington, DC, Apr. 2-5, 2013.

Bibliography

- [1] A. Oxenham. Forward masking: Adaptation or integration? *Journal of the Acoustical Society of America*, 109:732–741, 2001.
- [2] W. Jesteadt, S. Bacon, and J. Lehman. Forward masking as a function of frequency, masker level, and signal delay. *Journal of the Acoustical Society of America*, 71:950–962, 1982.
- [3] C. Lee, F. Soong, and K. Paliwal. *Automatic Speech and Speaker Recognition: Advanced Topics*. Springer, 1996.
- [4] J. S. Lim. *Speech Enhancement*. Prentice-Hall, 1983.
- [5] L. R. Rabiner and R. W. Schafer. *Digital Processing of Speech Signals*. Prentice-Hall, 1978.
- [6] H. Ding. *Speech Enhancement in Transform Domain*. PhD thesis, Nanyang Technological University, 2011.
- [7] X. Xiao. *Robust Speech Features and Acoustic Models for Speech Recognition*. PhD thesis, Nanyang Technological University, 2009.
- [8] S. Katagiri. *Handbook of neural networks for speech processing*. Artech House, 2000.
- [9] A. M. Kondo. *Digital speech: coding for low bit rate communication systems, 2nd edition*. Wiley, 2004.
- [10] T. F. Quatieri. *Discrete-time Speech Signal Processing: Principles and Practice*. Prentice Hall PTR, 2001.
- [11] C. Wei. *Noise Robust Signal Processing for Human Pitch Tracking and Bird Song Classification and Detection*. PhD thesis, University of California Los Angeles, 2012.

-
- [12] B. Strope. *Modeling Auditory Perception for Robust Speech Recognition*. PhD thesis, University of California Los Angeles, 1998.
 - [13] S. V. Vaseghi. *Advanced Digital Processing and Noise Reduction*. John Wiley, 2000.
 - [14] S. R. Quackenbush, T. P. Barnwell III, and M. A. Clements. *Objective Measures of Speech Quality*. Prentice Hall, 1988.
 - [15] B. J. Kybic. *Kalman Filtering and Speech Enhancement*. Ecole Polytechnique Federale De Lausanne, 1998.
 - [16] J. Droppo, A. Acero, and L. Deng. Evaluation of the SPLICE Algorithm on the Aurora 2 Database. In *Proc. Eurospeech Conference, International Speech Communication Association*, 2001.
 - [17] H. Li, B. Ma, and C. Lee. A Vector Space Modeling Approach to Spoken Language Identification. *IEEE Transactions on Audio, Speech, and Language Processing*, 15:271–284, 2007.
 - [18] B. Gold and N. Morgan. *Speech and audio signal processing - processing and perception of speech and music*. John Wiley and Sons, 1999.
 - [19] Y. Tsao and C. Lee. Two extensions to ensemble speaker and speaking environment modeling for robust automatic speech recognition. In *Automatic Speech Recognition & Understanding, 2007. ASRU. IEEE Workshop on*, pages 77–80, 2007.
 - [20] M. R. Sambur and N. S. Jayant. LPC analysis/synthesis from speech inputs containing quantizing noise or additive white noise. *IEEE Trans. Acoust., Speech, Signal Processing*, 24:488–494, 1976.
 - [21] B. H. Juang. Recent developments in speech recognition under adverse conditions. In *Proc. Int. Conf. Spoken Language Process*, pages 1113–1116, Kobe, Japan, November 1990.
 - [22] J. H. L. Hansen and M. A. Clements. Constrained iterative speech enhancement with application to speech recognition. *IEEE Trans. Signal Processing*, 39:795–805, 1991.
 - [23] L. Deng and X. Huang. Challenges in adopting speech recognition. *Communications of the ACM*, 47(1):69–75, 2004.
 - [24] Y. Gong. Speech recognition in noisy environments: A survey. *Speech Communication*, 16:261–291, 1995.

-
- [25] B. Juang. Speech recognition in adverse environments. *Computer Speech and Language*, 5:275–294, 1991.
 - [26] J. R. Deller, J. H. L. Hansen, and J. G. Proakis. *Discrete-time processing of speech Signals*. IEEE, 2000.
 - [27] C. Lee and Q. Huo. On adaptive decision rules and decision parameter adaptation for automatic speech recognition. *Proceedings of the IEEE*, 88(8):1241–1269, 2000.
 - [28] J. Gauvain and C. Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE Transactions on Speech and Audio Processing*, 2(2):291–298, 1994.
 - [29] M. Ostendorf, V. Digalakis, and O. Kimball. From HMM’s to segment models: a unified view of stochastic modeling for speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 4:360–378, 1996.
 - [30] O. Douglas. Automatic speech recognition: History, methods and challenges. *Pattern Recognition*, 41:2965–2979, 2008.
 - [31] J. Liu, Z. Wang, and X. Xiao. A hybrid SVM/DDBHMM decision fusion modeling for robust continuous digital speech recognition. *Pattern Recognition Letters*, 28:912–920, 2007.
 - [32] C. Leggetter and P. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *Computer Speech & Language*, 9(2):171–185, 1995.
 - [33] J. Li and C. Lee. Soft margin feature extraction for automatic speech recognition. In *INTERSPEECH*, pages 30–33, 2007.
 - [34] Y. Tsao and C. Lee. An ensemble modeling approach to joint characterization of speaker and speaking environments. In *Eurospeech ’07*, pages 1050–1053, 2007.
 - [35] P. J. Moreno. *Speech recognition in noisy environments*. PhD thesis, 1996.
 - [36] Y. Ephraim and D. Malah. Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator. *IEEE Trans. Acoust., Speech, Signal Processing*, 32:1109–1121, 1984.
 - [37] Y. Ephraim and D. Malah. Speech enhancement using a minimum mean-square error log-spectral amplitude estimator. *IEEE Trans. Acoust., Speech, Signal Processing*, 33:443–445, 1985.

-
- [38] S. F. Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Trans. Acoust., Speech, Signal Processing*, 27:113–120, 1979.
- [39] D. Yu, L. Deng, J. Droppo, J. Wu, Y. Gong, and A. Acero. Robust speech recognition using a cepstral minimum mean square error motivated noise suppressor. *IEEE Transactions on Audio, Speech and Language Processing*, 16:1061–1070, 2008.
- [40] J. Chen, J. Benesty, Y. Huang, and S. Doclo. New insights into the noise reduction Wiener filter. *IEEE Transactions on Audio, Speech and Language Processing*, 14:1218–1233, 2006.
- [41] X. Zhang, S.N. Koh, I.Y. Soon, and C. H. You. Post-processing in masking-based β -order MMSE speech enhancement. *Applied Acoustics*, 69:354–357, 2008.
- [42] M. K. Hasan, S. Salahuddin, and M. R. Khan. A modified a priori SNR for speech enhancement using spectral subtraction rules. *IEEE Signal Processing Letters*, 11(4):450–453, 2004.
- [43] I. Y. Soon and S. N. Koh. Speech enhancement using 2-D Fourier transform. *IEEE Trans. Speech and Audio Processing*, 11:717–724, 2003.
- [44] D. L. Donoho. De-noising by soft-thresholding. *IEEE Trans. Information Theory*, 41:613–627, 1995.
- [45] L. Deng, J. Droppo, and A. Acero. Estimating cepstrum of speech under the presence of noise using a joint prior of static and dynamic features. *IEEE Transactions on Audio, Speech and Language Processing*, 12:218–233, 2004.
- [46] X. Xiao, E. Chng, and H. Li. Normalization of the speech modulation spectra for robust speech recognition. *IEEE Transactions on Audio, Speech and Language Processing*, 16:1662–1674, 2008.
- [47] C. Chen and J. Bilmes. MVA Processing of Speech Features. *IEEE Transactions on Audio, Speech, and Language Processing*, 15:257–270, 2007.
- [48] P. Dai and I. Y. Soon. A temporal warped 2D psychoacoustic modeling for robust speech recognition system. *Speech Communication*, 53:229–241, 2010.
- [49] P. Dai, I. Y. Soon, and C. K. Yeo. 2D psychoacoustic filtering for robust speech recognition. In *Proceedings ICICS*, pages 1–5, 2009.

-
- [50] A. F. Erwood and C. S. Xydeas. Reduction of acoustic noise in speech signals by pre-processing and spectral subtraction. In *Proceedings of IERE International Conference on Digital Processing of Signals in Communications, Loughborough, England*, April 1981.
 - [51] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
 - [52] A. Sankar and C. Lee. A maximum-likelihood approach to stochastic matching for robust speech recognition. *Speech and Audio Processing, IEEE Transactions on*, 4(3):190–202, 1996.
 - [53] R. Lippmann. Review of neural networks for speech recognition. *Neural computation*, 1(1):1–38, 1989.
 - [54] B. Gold, N. Morgan, and D. Ellis. *Speech and audio signal processing: processing and perception of speech and music*. John Wiley & Sons, 2011.
 - [55] wikipedia. <http://en.wikipedia.org/>.
 - [56] R. Lippmann. An introduction to computing with neural nets. *IEEE ASSP Magazine*, 4(2):4–22, 1987.
 - [57] D. Morgan and C. Scofield. *Neural Networks and Speech Processing*. Springer, 1991.
 - [58] L. B. Thomas and A. Ravindran. Intelligibility enhancement of already noisy speech signals. *J. Audio Eng. Soc.*, 22:234–236, 1974.
 - [59] S. Young. *The HTK Book*. Cambridge University Engineering Department, 2006.
 - [60] L. Rabiner and B. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, 1993.
 - [61] R. Cox, C. Kamm, L. Rabiner, J. Schroeter, and J. Wilpon. Speech and language processing for next-millennium communications services. *Proc. IEEE*, 88(8):1314–1337, 2000.
 - [62] C.J. Weinstein. A linear prediction vocoder with voice excitation. In *Proceedings EASCON*, pages 30A–30G, 1975.
 - [63] L. Rabiner and M. Sambur. Voiced-unvoiced-silence detection using the Itakura LPC distance measure. In *Processings ICASSP*, volume 2, 1977.

-
- [64] A. V. Oppenheim and R. W. Schaffer. *Digital signal processing*. Prentice-Hall, 1975.
- [65] X. Cui and A. Alwan. Adaptation of childrens speech with limited data based on formant-like peak alignment. *Computer Speech & Language*, 20(4):400–419, 2006.
- [66] G. von Békésy. *Experiments in Hearing*. McGraw-Hill, 1960.
- [67] A. Acero. *Acoustical and environmental robustness in automatic speech recognition*. Springer, 1993.
- [68] F. Jelinek. *Statistical Method for Speech Recognition*. MIT Press, 1999.
- [69] F. Wessel, R. Schluter, K. Macherey, and H. Ney. Confidence measures for large vocabulary continuous speech recognition. *Speech and Audio Processing, IEEE Transactions on*, 9(3):288–298, 2001.
- [70] HTK website. <http://htk.eng.cam.ac.uk/>.
- [71] D. B. Dean, P. J. Lucey, S. Sridharan, and T. J. Wark. Fused HMM-adaptation of multi-stream HMMs for audio-visual speech recognition. *Interspeech, Antwerp, August*, pages 666–669, 2007.
- [72] K. M. Indrebo, R. J. Povinelli, and M. T. Johnson. Minimum Mean Squared Error Estimation of Mel-Frequency Cepstral Coefficients Using a Novel Distortion Model. *IEEE Transactions on Audio, Speech and Language Processing*, 16:1654–1661, 2008.
- [73] L. Deng, J. Wu, J. Droppo, and A. Acero. Analysis and comparison of two speech feature extraction/compensation algorithms. *IEEE Signal Processing Letters*, 12(6):477–480, 2005.
- [74] I. Y. Soon, S. N. Koh, and Z. X. Yuan. Selective magnitude subtraction for noise reduction. *Journal of Communications and Networks*, 2:226–229, 2000.
- [75] J. S. Lim and A. V. Oppenheim. Enhancement and bandwidth compression of noisy speech. *Proceedings of IEEE*, 67:1586–1604, 1979.
- [76] M. Kvale and C. Schreiner. Short term adaptation of auditory receptive fields to dynamic stimuli. *Journal of Neurophysiology*, 91:604–612, 2004.
- [77] X. Luo, I. Y. Soon, and C. K. Yeo. An auditory model for robust speech recognition. In *Proc. International Conference on Audio, Language and Image Processing (ICALIP)*, pages 1105–1109, April 2008.

-
- [78] S. A. Shamma. Speech processing in the auditory system II: Lateral inhibition and the central processing of speech evoked activity in the auditory nerve. *Journal of the Acoustic Society of America*, 78:1622–1632, 1985.
 - [79] H. Hermansky and N. Morgan. RASTA processing of speech. *IEEE Transactions on Audio, Speech and Language Processing*, 2:578–589, 1994.
 - [80] A. Acero and R. M. Stern. Robust speech recognition by normalization of the acoustic space. In *Proceedings ICASSP*, pages 893–896, 1991.
 - [81] Y. Wang, J. An, V. Sethu, and E. Ambikairajah. Perceptually motivated pre-filter for speech enhancement using Kalman filtering. In *Information, Communications & Signal Processing, 2007 6th International Conference on*, pages 1–4, 2007.
 - [82] S. Ou, X. Zhao, and J. Dong. Combining DCT and Adaptive KLT for Noisy Speech Enhancement. In *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on*, pages 2857–2860, 2007.
 - [83] J. H. Chang. Warped discrete cosine transform-based noisy speech enhancement. *IEEE Trans. Circuits and Systems II: Express Briefs*, 52(9):535–539, 2005.
 - [84] N. W. D. Evans, J. S. Mason, and M. J. Roach. Noise compensation using spectrogram morphological filtering. In *ProcEEDINGS 4th IASTED International Con. on Signal and Image Processing*, pages 157–161, 2002.
 - [85] A. Rezayee and S. Gazor. An adaptive KLT approach for speech enhancement. *IEEE Trans. Speech and Audio Processing*, 9(2):87–95, 2001.
 - [86] P. Scalart and J. Vieira Filho. Speech enhancement based on a a priori signal to noise estimation. In *Proceedings ICASSP*, volume 2, pages 629–632, 1996.
 - [87] M. Berouti, R. Schwartz, and J. Makhoul. Enhancement of speech corrupted by acoustic noise. In *IEEE Proc. ICASSP*, pages 208–211, April 1979.
 - [88] P. Lockwood and J. Boudy. Experiments with a nonlinear spectral subtractor (NSS), hidden Markov models and the projection, for robust speech recognition in cars. *Speech Communication*, 11(2-3):215–228, 1992.
 - [89] J. Lim. Evaluation of a correlation subtraction method for enhancing speech degraded by additive white noise. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 26(5):471–472, 1978.

-
- [90] I. S. Gradshteyn and I. M. Ryzhik. *Table of Integrals, Series, and Products*. Wiley, New York, 1980.
 - [91] R. M. Gray, A. Buzo, A. H. Gray Jr., and Y. Matsuyama. Distortion measures for speech processing. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-28:367–376, 1980.
 - [92] C. H. You, S. N. Koh, and R. Susanto. β -order MMSE spectral amplitude estimation for speech enhancement. *IEEE Trans. Speech and Audio Processing*, 13:457–486, 2005.
 - [93] Y. Ephraim and H. L. Van Trees. A signal subspace approach for speech enhancement. *IEEE Trans. Speech and Audio Processing*, 3:251–266, 1995.
 - [94] R. Hegger, H. Kantz, and L. Matassini. Noise reduction for human speech signals by local projections in embedding spaces. *IEEE Trans. Circuits and Systems I - Fundamental theory and applications*, 48:1454–1461, 2001.
 - [95] S. Gazor and A. Rezayee. An adaptive subspace approach for speech enhancement. In *Proceedings ICASSP*, volume 3, pages 1839–1842, 2000.
 - [96] U. Mittal and N. Phamdo. Signal/noise KLT based approach for enhancing speech degraded by colored noise. *IEEE Trans. Speech and Audio Processing*, 8:159–167, 2000.
 - [97] J. Allen. How do humans process and recognize speech? *IEEE Transactions on Acoustics, Speech and Signal Processing*, 2:567–577, 1994.
 - [98] H. Hermansky. Should recognizers have ears? *Speech Communication*, 25:3–27, 1998.
 - [99] D. M. Harris and P. Dallos. Forward masking of auditory nerve fiber responses. *Journal of Neurophysiology*, 42(4):1083–1107, 1979.
 - [100] S. Kramer and D. Teas. Forward masking of auditory nerve (n[_{sub 1}]) and brainstem (wave v) responses in humans. *The Journal of the Acoustical Society of America*, 72(3):795–803, 1982.
 - [101] I. S. Ryu, J. H. Ahn, and H. W. Lim. Evaluation of masking effects on speech perception in patients with unilateral chronic tinnitus using the hearing in noise test. *Otology & Neurotology*, 33(9):1472–1476, 2012.
 - [102] P. Dai and I. Y. Soon. A temporal frequency warped (TFW) 2D psychoacoustic filter for robust speech recognition system. *Speech Communication*, 54:402–413, 2012.

-
- [103] S. Rosen, P. Souza, C. Ekelund, and A. Majeed. Listening to speech in a background of other talkers: Effects of talker number and noise vocoding. *The Journal of the Acoustical Society of America*, 133(4):2431–2443, 2013.
 - [104] 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*, 28:357–366, 1980.
 - [105] B. Milner. A minimum-mean-square-error noise reduction algorithm on mel-frequency cepstra for robust speech recognition. In *Proceedings ICASSP*, pages 797–800, 2002.
 - [106] S. Shamma. Speech processing in the auditory system I: The representation of speech sounds in the responses of the auditory nerve. *Journal of the Acoustic Society of America*, 78:1612–1621, 1985.
 - [107] B. Stroppe and A. Alwan. A model of dynamic auditory perception and its application to robust word recognition. *IEEE Transactions on Speech and Audio Processing*, 5:451–464, 1997.
 - [108] J. Tchorz and B. Kollmeier. A model of auditory perception as front end for automation speech recognition. *The Journal of the Acoustical Society of America*, 106:2040–2050, 1999.
 - [109] K. Aikawa, H. Singer, H. Kawahara, and Y. Tohkura. A Dynamic Cepstrum Incorporating Time Frequency Masking and Its Application to Continuous Speech Recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 668–671, 1993.
 - [110] N. Virag. Single channel speech enhancement based on masking properties of the human auditory system. *IEEE Trans. Speech and Audio Processing*, 7:126–137, 1999.
 - [111] M. Holmberg, D. Gelbart, and W. Hemmert. Automatic speech recognition with an adaptation model motivated by auditory processing. *IEEE Trans. on Audio, Speech and Language Processing*, 14:43–49, 2006.
 - [112] B. Jiang, A. Liebl, P. Leistner, and J. Yang. Sound masking performance of time-reversed masker processed from the target speech. *Acta Acustica united with Acustica*, 98(1):135–141, 2012.
 - [113] Y. Hara, M. Tohyama, and K. Miyoshi. Effects of temporal and spectral factors of maskers on speech intelligibility. *Applied Acoustics*, 73(9):893–899, 2012.

-
- [114] R. Goldhor. *Representation of consonants in the peripheral auditory system: a modeling study of the correspondence between response properties and phonetic feature*. PhD thesis, 1985.
 - [115] J. Cohen. Application of an auditory model to speech recognition. *The Journal of the Acoustical Society of America*, 85(6):2623–2629, 1989.
 - [116] K. Park and S. Lee. An engineering model of the masking for the noise-robust speech recognition. *Neurocomputing*, 52-4:615–620, 2003.
 - [117] Y. M. Cheng and D. O’Shaughnessy. Speech enhancement based conceptually on auditory evidence. *IEEE Trans. Signal Processing*, 39:1943–1954, 1991.
 - [118] I. Brons, R. Houben, and W. Dreschler. Perceptual effects of noise reduction by time-frequency masking of noisy speech. *The Journal of the Acoustical Society of America*, 132(4):2690–2699, 2012.
 - [119] B. Zhang, Z. Long, and Y. Zhang. Effect of familiar masking background on target speech recognition. *Acta Acustica united with Acustica*, 98(2):328–333, 2012.
 - [120] T. Houtgast. Psychophysical Evidence for Lateral Inhibition in Hearing. *Journal of the Acoustical Society of America*, 51:1885–1894, 1972.
 - [121] K. Ishizuka and T. Nakatani. Noise robust voice activity detection based on periodic to aperiodic component ratio. *Speech Communication*, 52:43–49, 2010.
 - [122] L. Rabiner, M. Sambur, and C. Schmidt. Applications of a nonlinear smoothing algorithm to speech processing. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 23(6):552–557, 1975.
 - [123] C. Breithaupt, T. Gerkmann, and R. Martin. Cepstral smoothing of spectral filter gains for speech enhancement without musical noise. *Signal Processing Letters, IEEE*, 14(12):1036–1039, 2007.
 - [124] M. Brookes. *Voicebox*. <http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html>.
 - [125] J. Chen, T. Baer, and B. Moore. Effect of enhancement of spectral changes on speech intelligibility and clarity preferences for the hearing impaired. *The Journal of the Acoustical Society of America*, 131(4):2987–2998, 2012.
 - [126] A. Oxenham and C. Plack. Effects of masker frequency and duration in forward masking: further evidence for the influence of peripheral nonlinearity. *Hearing Research*, 150:258–266, 2000.

-
- [127] L. Lois. Backward and Forward Masking of Probe Tones of Different Frequencies. *The Journal of the Acoustical Society of America*, 34:1116–1117, 1962.
 - [128] I. Cohen. Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging. *IEEE Transactions on Speech and Audio Processing*, 11:466–475, 2003.
 - [129] R. Schluter and H. Ney. Using phase spectrum information for improved speech recognition performance. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, volume 1, pages 133–136 vol.1.
 - [130] P. S. Jafari, Kang Hou-Yong, Wang Xiaosong, Fu Qian-Jie, and Jiang Hui. Phase-sensitive speech enhancement for cochlear implant processing. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5104–5107.
 - [131] J. Jensen and H. L. Hansen. Speech enhancement using a constrained iterative sinusoidal model. *IEEE Trans. Speech and Audio Processing*, 9:731–740, 2001.
 - [132] R. Vetter. Single channel speech enhancement using MDL-based subspace approach in bark domain. In *Proceedings ICASSP*, volume 1, pages 641–644, 2001.
 - [133] P. Vary. Noise suppression by spectral magnitude estimation - mechanism and theoretical limits. *Signal Processing*, 8:387–400, 1985.
 - [134] European Telecommunications Standards Institute (ETSI). *ETSI ES 202 050 V1.1.5*. 2007.
 - [135] I. Y. Soon, S. N. Koh, and C. K. Yeo. Noisy speech enhancement using discrete cosine transform. *Speech Communication*, 24:249–257, 1998.
 - [136] M. Zilany M. K. Hasan and M. R. Khan. DCT speech enhancement with hard and soft thresholding criteria. *Electronics Letters*, 38(13):669–670, 2002.
 - [137] N. L. Phu, E. Ambikairajah, and E. Choi. An improved soft threshold method for DCT speech enhancement. In *2nd International Conference on Communications and Electronics*, pages 268–271, 2008.
 - [138] A. Papandreou, F. Hlawatsch, and G. Boudreaux-Bartels. Quadratic Time-Frequency Representations with Scale Covariance and Generalized Time-Shift Covariance: A Unified Framework for the Affine, Hyperbolic, and Power Classes. *Digital Signal Processing*, 8:3–48, 1998.

- [139] Y. Takahashi, T. Takatani, K. Osako, H. Saruwatari, and K. Shikano. Blind Spatial Subtraction Array for Speech Enhancement in Noisy Environment. *IEEE Transactions on Audio, Speech, and Language Processing*, 17:650–664, 2009.