Efficient Feature Extraction Based on Two-dimensional Cepstrum analysis for Speech Recognition

Hossein Marvi

Submitted for the Degree of Doctor of Philosophy from the University of Surrey

Centre for Vision, Speech and Signal Processing School of Electronics and Physical Sciences University of Surrey Guildford, Surrey GU2 7XH, U.K.

December 2004

© Hossein Marvi 2004
Summary

Solving speech recognition problems requires an adequate feature extraction technique to transform the raw speech signal to a set of feature vectors to preserve most of information corresponding to the speech signal. The features should ideally be compact, distinct and well representative of the speech signal. If the feature vectors do not represent the important content of the speech, the performance of the system will perform poorly regardless of the pattern recognition techniques applied. Many different feature extraction representations of the speech signal have been suggested and tried for speech recognition. The most popular features which are used currently are Mel-frequency cepstral coefficients (MFCC) and perceptual linear prediction (PLP), which are based on one dimensional cepstrum analysis.

The two dimensional cepstrum (TDC) is an alternative approach for time-frequency representation of any speech signal which can preserve both the instantaneous and transitional information of the speech signal. Here, in this thesis, the principle aim concerns the study of the two dimensional cepstrum analysis as a feature extraction technique for speech recognition. A novel feature extraction technique, two dimensional root cepstrum (TDRC) is also introduced. It has the advantage of an adjustable $\gamma$ parameter which can be used to optimise the feature extraction process, reducing the dimensions of the feature matrix and giving simple computation. In addition, the Mel TDRC has been proposed as a modified method of original TDRC to improve the accuracy. It is shown that both the TDC and the TDRC outperform the conventional cepstrum.

To preserve both magnitude and phase details of the speech signal simultaneously in a feature matrix, the Hartley transform (HT) is suggested as a substitute for the Fourier transform (FT) in two-dimensional cepstrum analysis. Experimental results demonstrate the enhanced capability of the HT in the two dimensional root cepstral analysis to improve recognition accuracy. An experimental comparative study of 9 kinds of feature extraction methods based on cepstral analysis are also carried out.

Key words: Two dimensional cepstrum, Root cepstrum, Feature extraction, Speech recognition, Hartley transform
Acknowledgements

I would like to express my deepest appreciation and gratitude to my supervisor Doctor Ted Chilton for his invaluable guidance and encouragement.

I would also like to thank all the members of the Centre for Vision Speech and Signal Processing, for providing a nice and friendly environment.

During this course, I was financially supported by the Ministry of Science, Research and Technology in Iran.

Finally, I would like to thank my wife, Marzieh, and my Daughter Hanieh, My son Mohammed, for their support and patience during my study.
Publication

Some parts of this thesis have been published as follows:


Contents

1 Introduction .................................................. 1
  1.1 Motivation and aim ..................................... 2
  1.2 Structure of this thesis ................................. 2
  1.3 Original Contributions ................................. 4

2 General issues in speech recognition ................. 7
  2.1 Introduction ............................................. 7
  2.2 Main stages of speech recognition system .......... 8
    2.2.1 Pre-processing of speech ......................... 8
    2.2.2 Feature extraction ................................. 8
    2.2.3 Pattern recogniser for speech recognition ...... 9
    2.2.4 Post processing and Decision maker ............ 11
  2.3 Time alignment and Normalisation .................. 11
    2.3.1 Linear time alignment ............................ 12
    2.3.2 Non-Linear time alignment ....................... 13
  2.4 Types of speech recognition system ................ 13
    2.4.1 Isolated word and connected word recognition .. 13
    2.4.2 Speaker independent and dependent recognition system 14
  2.5 Summary ................................................. 14

3 Overview of current feature extraction methods ..... 15
  3.1 Introduction ............................................. 15
  3.2 Ideal feature extraction ............................... 16
  3.3 Cepstral analysis ...................................... 18
    3.3.1 Real Cepstrum ...................................... 19
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3.2</td>
<td>Complex cepstrum</td>
<td>22</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Differential cepstrum</td>
<td>23</td>
</tr>
<tr>
<td>3.3.4</td>
<td>Root cepstrum</td>
<td>24</td>
</tr>
<tr>
<td>3.4</td>
<td>Mel-Frequency cepstral coefficients</td>
<td>25</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Pre-emphasis</td>
<td>26</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Windowing and Spectral analysis</td>
<td>28</td>
</tr>
<tr>
<td>3.4.3</td>
<td>Filter bank analysis</td>
<td>29</td>
</tr>
<tr>
<td>3.4.4</td>
<td>Logarithm</td>
<td>30</td>
</tr>
<tr>
<td>3.4.5</td>
<td>Discrete cosine transform</td>
<td>30</td>
</tr>
<tr>
<td>3.5</td>
<td>Linear prediction analysis</td>
<td>31</td>
</tr>
<tr>
<td>3.5.1</td>
<td>Cepstral Coefficients derived from LPC</td>
<td>33</td>
</tr>
<tr>
<td>3.5.2</td>
<td>Comparison between LPC-cepstrum and FFT-cepstrum</td>
<td>34</td>
</tr>
<tr>
<td>3.6</td>
<td>Line spectral frequencies</td>
<td>35</td>
</tr>
<tr>
<td>3.6.1</td>
<td>Relationship between the LPC cepstrum and LSF</td>
<td>36</td>
</tr>
<tr>
<td>3.6.2</td>
<td>LSF for speech recognition</td>
<td>36</td>
</tr>
<tr>
<td>3.7</td>
<td>Perceptual linear prediction representation</td>
<td>37</td>
</tr>
<tr>
<td>3.7.1</td>
<td>Spectral analysis</td>
<td>37</td>
</tr>
<tr>
<td>3.7.2</td>
<td>Critical band analysis</td>
<td>37</td>
</tr>
<tr>
<td>3.7.3</td>
<td>Equal loudness pre-emphasis</td>
<td>40</td>
</tr>
<tr>
<td>3.7.4</td>
<td>Intensity-Loudness power law</td>
<td>40</td>
</tr>
<tr>
<td>3.7.5</td>
<td>Autoregressive modelling</td>
<td>41</td>
</tr>
<tr>
<td>3.7.6</td>
<td>Post-processing</td>
<td>41</td>
</tr>
<tr>
<td>3.8</td>
<td>Relative spectral processing of speech</td>
<td>41</td>
</tr>
<tr>
<td>3.8.1</td>
<td>RASTA-PLP technique</td>
<td>42</td>
</tr>
<tr>
<td>3.9</td>
<td>Feature Normalisation</td>
<td>44</td>
</tr>
<tr>
<td>3.9.1</td>
<td>Cepstral mean normalisation</td>
<td>44</td>
</tr>
<tr>
<td>3.10</td>
<td>Dynamic features</td>
<td>44</td>
</tr>
<tr>
<td>3.10.1</td>
<td>Delta cepstrum</td>
<td>45</td>
</tr>
<tr>
<td>3.10.2</td>
<td>Delta-Delta cepstrum</td>
<td>46</td>
</tr>
<tr>
<td>3.11</td>
<td>Summary and discussion</td>
<td>46</td>
</tr>
</tbody>
</table>
# Contents

## 4 Two-dimensional cepstrum analysis

4.1 Introduction ................................................. 49

4.2 Why Two-dimensional features? ....................... 50

4.3 Approaches to two dimensional cepstrum analysis .... 51

4.3.1 Cepstrum of two dimensional signal .......... 51

4.3.2 Cepstrum time matrices ......................... 52

4.3.3 Two dimensional cepstrum ......................... 53

4.3.4 Schemes for computing the TDC ................. 56

4.3.5 Comparison between TDC and CTM ............. 57

4.4 Regions of TDC ............................................. 58

4.5 Two-dimensional root cepstrum analysis ............. 60

4.5.1 Definition of TDRC ................................. 60

4.5.2 Properties of TDRC ................................. 64

4.6 Complex TDRC ............................................. 67

4.7 Conclusion .................................................... 68

## 5 Evaluation of the performance of Two dimensional cepstrums

5.1 Introduction ................................................ 71

5.2 Experimental arrangement .............................. 72

5.3 Classification .............................................. 73

5.3.1 Distance classification derived from the Bayes's rule .. 74

5.3.2 Other types of distance classifier ................. 75

5.4 Estimation of the Density Function of Features ........ 75

5.4.1 Correlation matrix for two dimensional features ..... 77

5.5 Experiments on the TDC ............................... 80

5.5.1 TDC performance .................................... 81

5.5.2 TDC feature matrix dimensions ................ 83

5.5.3 Optimum frame length .............................. 87

5.5.4 Classifier performance comparison .............. 88

5.5.5 Comparison of ODC and TDC ................... 88

5.6 Experiments on the TDRC .............................. 90

5.6.1 TDRC feature matrix dimensions ............... 90
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.6.2</td>
<td>Optimum frame length</td>
<td>91</td>
</tr>
<tr>
<td>5.6.3</td>
<td>Effect of $\gamma$ variation</td>
<td>93</td>
</tr>
<tr>
<td>5.6.4</td>
<td>A comment on performance optimisation</td>
<td>94</td>
</tr>
<tr>
<td>5.6.5</td>
<td>Classifier performance comparison</td>
<td>95</td>
</tr>
<tr>
<td>5.7</td>
<td>Noisy speech recognition</td>
<td>95</td>
</tr>
<tr>
<td>5.8</td>
<td>Comparison of TDC and TDRC</td>
<td>98</td>
</tr>
<tr>
<td>5.9</td>
<td>Gaussian mixture model for features</td>
<td>100</td>
</tr>
<tr>
<td>5.9.1</td>
<td>GMM parameters estimation</td>
<td>101</td>
</tr>
<tr>
<td>5.9.2</td>
<td>Initialisation of GMM</td>
<td>103</td>
</tr>
<tr>
<td>5.9.3</td>
<td>Experiments with GMM</td>
<td>104</td>
</tr>
<tr>
<td>5.10</td>
<td>Summary and conclusion</td>
<td>106</td>
</tr>
<tr>
<td>6</td>
<td>Developing the proposed methods</td>
<td>109</td>
</tr>
<tr>
<td>6.1</td>
<td>Introduction</td>
<td>109</td>
</tr>
<tr>
<td>6.2</td>
<td>Main Point of LDA</td>
<td>110</td>
</tr>
<tr>
<td>6.3</td>
<td>Experiments with LDA</td>
<td>111</td>
</tr>
<tr>
<td>6.3.1</td>
<td>TDC with LDA</td>
<td>111</td>
</tr>
<tr>
<td>6.3.2</td>
<td>TDRC with LDA</td>
<td>112</td>
</tr>
<tr>
<td>6.4</td>
<td>Mel-scale frequency TDRC</td>
<td>115</td>
</tr>
<tr>
<td>6.5</td>
<td>Experiments with Mel TDC and Mel TDRC</td>
<td>116</td>
</tr>
<tr>
<td>6.5.1</td>
<td>Experiments with MTDC</td>
<td>116</td>
</tr>
<tr>
<td>6.5.2</td>
<td>Experiments with MTDRC</td>
<td>116</td>
</tr>
<tr>
<td>6.6</td>
<td>Hartley Transform</td>
<td>118</td>
</tr>
<tr>
<td>6.6.1</td>
<td>Two-dimensional Hartley transform</td>
<td>120</td>
</tr>
<tr>
<td>6.6.2</td>
<td>Hartley cepstrum</td>
<td>121</td>
</tr>
<tr>
<td>6.7</td>
<td>Hartley Two-dimensional Root Cepstrum</td>
<td>122</td>
</tr>
<tr>
<td>6.7.1</td>
<td>Experiments with HTDRC</td>
<td>123</td>
</tr>
<tr>
<td>6.8</td>
<td>A comparison of feature extraction methods</td>
<td>124</td>
</tr>
<tr>
<td>6.8.1</td>
<td>Names of all used methods</td>
<td>125</td>
</tr>
<tr>
<td>6.8.2</td>
<td>Comparative criteria</td>
<td>126</td>
</tr>
<tr>
<td>6.8.3</td>
<td>Experiments</td>
<td>126</td>
</tr>
<tr>
<td>6.9</td>
<td>Summary and conclusion</td>
<td>128</td>
</tr>
</tbody>
</table>
## Contents

7 Conclusion and future work 131
  7.1 Summary of the thesis ............................................... 131
  7.2 Future work .............................................................. 133

A Details of Experimental Methodology 137
  A.1 General experimental set up ....................................... 137
  A.2 Training process ...................................................... 138
  A.3 Testing process ....................................................... 139
  A.4 Recognition process ................................................ 139

B Statistical significance test for comparing two classifiers 141
Contents
List of Figures

2.1 Block diagram of speech recognition system .................................................... 8
2.2 A multi layer perceptron .................................................................................. 12

3.1 Block diagram for computing the Cepstrum ................................................... 19
3.2 The process of lifting in cepstrum analysis ..................................................... 21
3.3 Analysing a frame of speech waveform to obtain cepstrum and spectral envelope ............................................................................................................. 22
3.4 The real cepstrum for a frame of voiced speech ............................................. 23
3.5 Homomorphic deconvolution .......................................................................... 24
3.6 Pseudo-Cepstral analysis of a speech frame .................................................. 26
3.7 Computing mel-frequency cepstrum coefficients ........................................... 27
3.8 Pre-emphasis .................................................................................................... 28
3.9 The Mel Scale .................................................................................................. 29
3.10 Filters for generating mel-frequency cepstrum coefficients .......................... 30
3.11 A general discrete time model for speech production .................................... 32
3.12 Perceptual linear prediction representation of speech signal ....................... 38
3.13 Bark-scale warping function ......................................................................... 39
3.14 Critical-band basic functions ......................................................................... 40
3.15 Block diagram of RASTA-PLP speech processing ...................................... 42

4.1 3-D representation of TDC matrix for digits: one, two, three and four ........ 54
4.2 3-D representation of TDC matrix for digits: five, six, seven and eight ........................................... 55
4.3 Regions of two-dimensional cepstrum ............................................................... 59
4.4 Two dimensional root cepstrum analysis of speech signal ............................. 61
4.5 Time-frequency representation of word “She” using the TDRC features ........ 63
4.6 The three first columns and rows of the TDRC matrix .................................. 65
List of Figures

4.7 Regions of two-dimensional root cepstrum .......................................................... 66
4.8 The compression effects of $\gamma$ along 'u' and 'v' axis ..................................... 67
4.9 Complex TDRC .................................................................................................. 68

5.1 Histogram for features produced by TDC method .......................................... 77
5.2 Histogram for features produced by TDRC method ......................................... 79
5.3 Correlation matrix for TDC ................................................................................ 80
5.4 Correlation matrix for TDRC ............................................................................. 81
5.5 The performance of ODC and TDC methods on recognition accuracy ............. 82
5.6 The Effect of changing column number and row number on recognition accuracy before and after removing the DC-offset .................................................. 86
5.7 Comparison of the effects of feature selection by row and column changing on recognition accuracy .......................................................... 87
5.8 The effect of increasing frame length (TDC method) ....................................... 88
5.9 A comparison of recognition accuracy for four classifiers (TDC method) ......... 89
5.10 Effects of row and column changing on recognition accuracy for TDRC method ........................................................................................................ 92
5.11 The effect of increasing frame length (TDRC method) .................................... 92
5.12 The effects of $\gamma$ variation when $\gamma$ is less than $-1$ (left) and more than 1 (right) .......................................................... 94
5.13 The effects of $\gamma$ variation when $-1 < \gamma < 0$, left, and $0 < \gamma < 1$, right .... 94
5.14 Effect of changing $\gamma$ on recognition accuracy. ............................................. 95
5.15 Classifier performance comparison for TDRC features .................................. 96
5.16 Results of Recognition accuracy in noisy conditions ....................................... 97
5.17 Results of comparison of TDC and TDRC methods. ....................................... 98
5.18 Depiction an M component Gaussian mixture model ..................................... 101
5.19 Gaussian Mixture approximation for the first feature of feature vector as an example. .......................................................... 105

6.1 Results of recognition accuracy (TDRC method) before and after LDA .......... 114
6.2 Comparison of using Mel TDRC and TDRC on recognition accuracy .......... 118
6.3 The Fourier and the Hartley transform of a frame of voiced speech signal .... 120
6.4 Block diagram of Two-dimensional Hartley Root Cepstrum analysis .......... 123
6.5 Recognition accuracy as a function of $\gamma$ variation ..................................... 125
List of Figures

6.6 Separability Criteria using different types of features extraction methods 128
6.7 Separability Criteria of one and Two-dimensional methods ............ 129
6.8 Recognition accuracy for One-dimensional and two-dimensional methods 130
List of Tables

5.1 Kurtosis of Features both TDC and TDRC methods ........................................ 78
5.2 Recognition accuracy using ODC and TDC method (in %) ........................... 82
5.3 The effect of changing the number of rows on Recognition accuracy .......... 84
5.4 The effects of changing number of columns on Recognition accuracy ......... 85
5.5 The Statistical analysis for column adding .................................................... 86
5.6 Recognition accuracy (in %) for various feature numbers and classifier using ODC .................................................. 89
5.7 Recognition accuracy (in %) for various feature numbers and classifier using TDC .................................................. 90
5.8 The Statistical analysis of a comparison of TDC and TDRC in noisy conditions .................................................. 97
5.9 Recognition accuracy (in %) for various feature numbers and classifier using TDC (repeated of table 5.7) .................. 99
5.10 Recognition accuracy (in %) for various feature numbers and classifier using TDRC .................................................. 99
5.11 Mean square results ..................................................................................... 106
5.12 GMM results ................................................................................................ 107

6.1 The results of TDC with and without LDA .................................................. 112
6.2 The results of TDRC with LDA .................................................................... 113
6.3 Statistical analysis of comparison of TDC and TDRC with and without LDA .................................................. 114
6.4 The results of TDC and Mel TDC .................................................................. 117
6.5 The results of Mel TDRC .............................................................................. 117
6.6 The results of HTDRC .................................................................................. 124
6.7 Recognition accuracy (%) for all types of methods ......................................... 129

A.1 List of words .................................................................................................. 138
Human communication is complex and varied. Speech is a most natural and efficient
way for communication and to exchange information between human beings. It is a
natural and universal form of communication that is a great deal faster than typing or
writing.

Speech is a sequence of acoustic waves which are transmitted over time through a
medium, air. Speech sounds are produced as a result of acoustic vibrations when breath
is exhaled from the lungs and causes either vibration of the vocal cords or turbulence
at some point of constriction in the vocal tract. Speech is perceived by the inner ear
in humans. It activates oscillations of small elements in the media of the inner ear and
these oscillations are transmitted to a specific part of brain for further processing.

Speech can be modelled as the convolution of an excitation source, either voiced or
unvoiced, with the vocal tract filter impulse response.

Since spoken communication is so fundamental to human beings, its understanding
and the process of recognition by machine has been considered and developed over
many years. To make a real intelligent computer, it is important that the machine
can hear, understand and act upon spoken information and also speak to complete the
information exchange. It follows, therefore, that automatic speech recognition (ASR)
is a very important and useful facility for some types of machine to possess.

Speech recognition is the process of automatically extracting and determining linguistic
information conveyed by a speech wave using a machine. The aim of speech recognition
is to map a speech signal to a corresponding sequence of linguistic units which it represents. To perform this, a series of acoustic features should be extracted from the speech signal, and then pattern recognition techniques need to be used to classify the patterns.

1.1 Motivation and aim

Solving speech recognition problems require adequate feature extraction to transform the raw speech signal into a set of features. This transformation reduces the data and makes the recognition easier through dealing with smaller amounts of data. If the feature vectors do not represent the important content of the speech, the system will perform poorly regardless of which pattern recognition techniques are applied. During past years, extensive research has been carried out on the extraction of suitable features for speech recognition. Most methods are based on the one dimensional cepstrum while little work has been done using the two dimensional cepstrum. The two dimensional cepstrum is an approach for time-frequency representation of any speech signal which can preserve both the instantaneous and transitional information. Hence, in this thesis, the principle aim concerns the study of the two dimensional cepstrum as a feature extraction technique for speech recognition. The problem has been restricted to isolated word recognition in the first instance. Furthermore the two dimensional root cepstrum is introduced as a novel method of feature extraction. In addition, the Mel TDRC has been proposed as a modified method of original TDRC to improve the accuracy. Alternative approaches concern the enhanced capability of the Hartley transform (HT) in two dimensional root cepstral analysis to improve recognition accuracy.

1.2 Structure of this thesis

The work presented in this thesis is organised as follows: Chapter 2 presents a general discussion of some aspects of speech recognition. First, the main stages of a speech recognition system are presented including pre-processing, feature extraction, pattern recognition and post processing. Since an utterance spoken
1.2. Structure of this thesis

by different speakers or the same speaker at different times has different time duration it is necessary in conventional feature extraction methods to perform time normalisation or time alignment between reference and test patterns. The two main methods used are Linear Time Alignment and Non-Linear Time Alignment which are described briefly in chapter 2. The next issue, which is discussed in chapter 2, are the different types of speech recognition system: isolated word or connected word recognition and speaker independent or dependent recognition systems.

Chapter 3 reviews different conventional approaches which are used for feature extraction methods in current speech recognition systems. Most of these methods use cepstral analysis as the main part of the feature extraction process. Thus, several cepstral analysis techniques, including the mel frequency cepstrum, are discussed. LPC is another dominant type of speech representation and is discussed briefly while Perceptual linear prediction (PLP), another popular speech feature extraction technique, is also presented in this chapter. Moreover, the feature normalisation and dynamic features are discussed which is necessary for refining conventional features.

The time-frequency or two dimensional cepstrum is introduced in chapter 4. As most feature extraction methods involve a one dimensional feature either time or frequency they omit important features for recognition which exist in both time and frequency. The two dimensional cepstrum (either TDC or TDRC) can simultaneously preserve various different types of information contained in the speech signal. Different types of two dimensional cepstrum are illustrated in this chapter. To overcome the weakness of original two-dimensional cepstrum we introduce the two-dimensional root cepstrum (TDRC) in this chapter. The properties of TDRC are also discussed in details and some initial experiments are carried out to support the theoretical discussion.

Although some primary experiments on the performance of the two dimensional cepstrum are given in chapter 4, the performance of the suggested methods are evaluated experimentally as efficient and simple methods for speech recognition in chapter 5. It is shown that both the TDC and the TDRC outperform the conventional cepstrum. It is demonstrated that the TDRC outperforms the TDC while presenting significant reduction in the size of the feature matrix due to efficient compression ability the TDRC.
Chapter 1. Introduction

The superior performance of TDRC over TDC on noisy speech is also discussed in this chapter. In the penultimate section of this chapter, the GMM is suggested to estimate the pdf of two dimensional features. Although it improves the results, we will see that there are some difficulties in implementation.

In chapter 6 the proposed feature extraction methods described in chapter 4 and 5 are developed. As the first step, Linear discriminant analysis is applied to the features to achieve extra dimensionality reduction while preserving as much of the class discriminatory information as possible and improving the recognition accuracy. Furthermore, we will introduce the Mel TDRC as a modified method of the original TDRC to improve the accuracy. To preserve both magnitude and phase details of the speech signal simultaneously in the feature matrix, the Hartley transform (HT) is suggested as a substitute for the FT in two-dimensional cepstrum analysis.

An experimental comparative study of 9 kinds of feature extraction methods for isolated word speech recognition are also carried out in this chapter. The experiments are conducted on the TIMIT data base. To compare the methods, two measures are used, separability criteria and recognition accuracy. Finally a summary of this thesis and suggested future research work are given in chapter 7.

1.3 Original Contributions

The main contributions of this thesis are summarised as follows:

- We propose the two-dimensional methods for isolated word speech recognition. In this method a feature matrix is considered for a whole word instead of using feature vectors for each frame [57, 55]. Furthermore recognition time is short in this method because the words are recognised based on computing the distance between the reference and test feature matrix.

- A comparison study of different distance measures based on TDC features for speech recognition has been illustrated [60].

- We introduce the two-dimensional root cepstrum as a novel method of feature extraction[17]. This method has some advantages such as an adjustable $\gamma$ param-
1.3. Original Contributions

eter which can be used to optimise the feature extraction process and reduces the dimensions of the feature matrix.

- We applied the LDA to TDRC features to achieve extra dimensionality reduction while preserving as much of the class discriminatory information as possible and thus improving the recognition accuracy [56].

- We introduce the Mel TDRC which produces much better results than the original TDC and TDRC.

- We suggest the idea of substitution of the HT for the FT in TDRC to preserve both magnitude and phase details of speech signal simultaneously.

- We show that the TDRC outperforms the original TDC for noisy speech recognition [58].

- We show that the TDRC method is superior to the ODC methods for speaker independent speech recognition [59].

- The performance of 4 types of one dimensional cepstrum methods are compared with 5 types of 2D method (total 9) to find the best method. This comparison has been done using two different criteria, separability criteria and recognition accuracy.
Chapter 2

General issues in speech recognition

2.1 Introduction

Speech recognition by machine is a complex task due to the great variety of speech signals. It has been the subject of much research over the past few decades. The speech signal varies for a given word both between speakers and for multiple utterances by the same speaker. Speech contains information encoded in a signal that is generated by a speaker. As we mentioned in chapter one, the task of a speech recognition system is to extract the encoded information from speech which requires an efficient feature extraction method. The principle aim of this work concerns feature extraction, but before starting to give more details about feature extraction methods, it is necessary to describe some general aspects of speech recognition systems.

In this chapter some general issues concerning speech recognition will be outlined. In the first part, after the introduction, section (2.2), the main stages of a speech recognition system will be described. In section (2.3), two main methods of time alignment, linear time alignment and Non-Linear time alignment are described briefly. The next issue, which is discussed in section (2.4), concerns the types of speech recognition system. It can be consider from different points of view such as: Isolated word or connected
word recognition and Speaker independent or dependent recognition system. Finally a summary of this chapter are given in section (2.5).

2.2 Main stages of speech recognition system

A speech recognition system can be divided into four stages as shown in Fig.2.1. These stage are: pre-processing of speech, feature extraction, pattern recognition system, and post-processing. As can be seen from this figure after the pre-processing of the speech, the features extracted from the speech are compared to previously stored patterns in the classification stage. The model may represent either a word or a smaller sub-word unit such as a phoneme. The model with the closest match to the input utterance is chosen as the recognised result. In the following section these stages are described briefly:

![Figure 2.1: Block diagram of speech recognition system](image)

2.2.1 Pre-processing of speech

The first stage of speech recognition is pre-processing the speech. This stage include an analogue to digital converter, filtering to remove the noise effects and pre-emphasise. Many recogniser systems pre-emphasise the speech signal before feature extraction so that high and low frequencies receive equal weight in feature extraction stage [105].

2.2.2 Feature extraction

Feature extraction is a process to convert the speech signal into a suitable parametric form which gives a good representation of the speech. The goal of the feature extraction is to incorporate, into the system, relevant information for classification i.e removing
the redundant information and compress the data as much as possible. A good feature extraction of patterns simplifies significantly the recogniser task. It is well known that the performance of a speech recognition system depends heavily on the types of features which are used in the system. The most commonly used parameters to date are based on the one dimensional cepstral analysis, such as Mel-frequency cepstral coefficients (MFCC) and perceptual linear prediction (PLP). An overview of different approaches which are used as feature extraction methods in current speech recognition systems will be discussed in the next chapter.

2.2.3 Pattern recogniser for speech recognition

The purpose of pattern recognition or pattern matching techniques is to determine to which class, a given input sample belongs. This is based on a set of selected features extracted from the pattern which make up the feature vector. In the context of speech recognition the aim of pattern recognition is to recognise the speech based on its previous reference pattern of speech which are extracted from the speech signal. The classifier uses these features to assign an input object to the correct class. Pattern recognition for speech falls into three main categories: Classical statistical pattern recognition, Hidden Markov model, and Artificial Neural Network.

Classical statistical pattern recogniser

In classical statistical pattern recognition, statistical information obtained from features, which are extracted from a known set of patterns (the training set) in the feature extraction stage, are used to determine suitable parameters to represent each pattern class. The feature vector is a random variable and its density function depends on the original data and its class.

There are two approaches for a statistical pattern recogniser.

1. A template matching approach which involves the features directly. A template matching involves a comparison of an average of features, computed on the test data, to a collection of stored averages for each of the classes in training. A special
case of this approach is simple template matching, where the whole pattern is compared with a reference pattern by measuring the distance between feature means.

2. Probabilities or likelihood methods which perform a comparison of probabilistic methods of features. Probabilistic modelling refers to modelling each classes by probability distributions and classification decisions are based on probabilities or likelihoods of a sample and model distribution consistency. Thus in this case the most important problem is the estimation of density function [110].

Hidden Markov model

Hidden Markov model (HMM) is the popular pattern recogniser which can be applied to many kinds of pattern recognition problem, especially in speech recognition. The theory of HMM was published in a series of paper by Baum and Jelinek et al. It was applied to speech processing by Baker and Jelinek [89, 40, 8]. The use of HMM is described in a number of tutorial papers of which the one by Rabiner is well known [91].

The underlying idea in this approach is to model the speech signal as a set of definite states. Each state has its own probability distribution and a set of state transition probabilities which model the behaviour of the signal in time. As a result, every possible state sequence can be assigned a probability of occurrence. Since their introduction, HMM have been developed extensively and have been proven to be one of the most successful approaches to speech recognition to date.

Artificial Neural Network

Artificial neural networks (ANN) are massively parallel processing systems made up of a number of very simple processing elements called neurons. These neurons are interconnected by links with adjustable weights in an attempt to model biological neural networks [50]. The ANN was initially proposed to simulate the problem solving procedure of human brain. ANN has been used in many classification application and
2.3. *Time alignment and Normalisation*

Pattern recognitions. It has shown some success for speech pattern classification tasks [97]. Instead of generating a template for each recognition unit, ANN approaches model the recognition unit by the pattern activity in interconnected nodes by learning strategies. There are many different types of neural networks, based on a variety of different algorithms and topologies. In the field of speech recognition, the multi-layer perception and Kohonen's self-organising feature maps have emerged as being the most successful. Fig. 2.2 shows a multi-layer perception with two hidden layers. The input signal propagates layer-by-layer from left to right. At each neuron the input values are multiplied with weights and summed up to form a network which is passed through a nonlinear function to form the output layer. The training process in ANN often requires many iterations over large amounts of training data which can be difficult and expensive in some applications.

A clear distinction should be drawn between HMM and ANN techniques. In the case of the HMM, it belongs to a group of classifiers called generative models. Generative models are those where the joint probability, \( p(x, y) \), of feature input vector \( x \) and label \( y \) are modelled from training data. Bayes' rule is then used to find the conditional probability, \( p(y|x) \). These models are relatively easy to train and need less data than discriminative models. On the other hand, ANNs are described as discriminative models where the conditional probability \( p(y|x) \) is modelled directly from the training data. These models require more data for training and are harder to train [115, 95]. Also, some researches have tried to combine ANN with HMM modelling to form a hybrid HMM-ANN system for speech recognition [11, 97].

2.2.4 Post processing and Decision maker

In this stage the reference pattern similarity scores are used to decide which reference pattern best matches the unknown test pattern to make the final decision.

2.3 Time alignment and Normalisation

One of the important problems in comparing a test signal and a template signal is that they should be at the same length. But this does not happen because, in speech, the
same word spoken by different speakers or the same speaker at different times have different time durations. There are two main methods to solve this problem, linear time alignment and non-linear time alignment.

2.3.1 Linear time alignment

Linear time alignment algorithms are the simplest algorithms to implement and they can be used for both expansion or compression of the speech pattern vector. There are various ways of implementing linear algorithms but all use the basic method of deleting feature vectors to shorten the speech pattern and duplicating feature vectors to lengthen the speech pattern. An example is to duplicate or delete vectors at regular intervals along the pattern vector until the speech pattern is the correct size. However, they take no account of the importance of the feature vectors within the pattern vector and important features, therefore, may be lost in the process. This method has the advantage of simplicity in comparison with non-linear method.
2.3.2 Non-Linear time alignment

Non-linear time alignment algorithms are more complicated and involve higher computational expenditure. In this method, an input template is non-linearly stretched or compressed to make match to the reference template. In other words, this technique allows parts of feature pattern to be stretched or compressed differently than other parts of the feature pattern. Dynamic time warping (DTW) is a popular method in this area [89]. It is an earlier and still widely used pattern matching technique for speech recognition. Although the HMM and DTW are apparently two different techniques for speech recognition, it has been shown that there is a close relationship between them. The DTW can be considered as a special case of HMM [19][41].

2.4 Types of speech recognition system

Speech recognition systems can be divided from a different point of view. The first is divided into isolated word and connected word recognition. The next approach is divided to speaker independent and dependent recognition system.

2.4.1 Isolated word and connected word recognition

In isolated word recognition the aim is to recognise the discrete utterance thus the utterance or words are separated by a pause. In contrast in connected or continuous word recognition the aim is to recognise the word boundaries which are typically not detectable. It is obvious that continuous speech recognition is considerably more difficult than isolated word recognition. Although current research on continuous speech recognition is more predominant and has been developed for many real-world applications, there are still markets where the application of isolated word recognition systems is appropriate such as recognising the telephone numbers, spelled number and address, command and control applications in which the user is required to speak the command words one at a time.
2.4.2 Speaker independent and dependent recognition system

A speaker independent speech recognition system should respond successfully to any speaker without having been specifically trained on that speaker. On the other hand, speaker dependent systems are trained on one speaker or some speakers and, therefore, are intended for use with that speaker or those speakers from the training. For a given speech recognition task, a speaker dependent system normally performs better than a speaker independent system and needs less computation but it requires more storage. In practice, some applications, such as personal voice typewriters and data access from a personal computer, can be speaker dependent, while others, say air ticket reservation and bank balance inquiry must be speaker independent.

2.5 Summary

In this chapter some general aspects of speech recognition system were discussed. First the main stages of a speech recognition system were presented which are per-processing, feature extraction, pattern recognition and post processing.

The most important part of a speech recognition system is feature extraction which makes the next steps easier. It is a well known fact that the performance of any speech recognition system depends heavily on feature extraction. An overview of current feature extraction methods are given in the next chapter. Since an utterance spoken by different speakers or the same speaker at different times has different time duration, it is necessary to do time normalisation or time alignment between reference and test pattern in classical method. We will see in chapter 4 that the TDC method does not need a time normalisation. The two main methods are linear time alignment and non-linear time alignment which are described briefly in this chapter. The next issue, which was discussed in this chapter, is the types of speech recognition. It can be considered from different point of view such as: isolated word or connected word recognition and speaker independent or dependent recognition system.
Chapter 3

Overview of current feature extraction methods

3.1 Introduction

Feature extraction is a process that computes certain feature vectors from the speech signal usually by short-term spectral analysis techniques. In the context of speech recognition, the main goal of feature extraction is to compute a sequence of feature vectors providing a compact representation of the given speech signal while retaining the salient features required. For each short frame of speech (e.g., 20 ms) the extraction procedure outputs a feature vector that describes the acoustical characteristics of this frame. Feature extraction has a great influence on the efficiency of recognition. In other words, the identification of a robust feature for recognition is an important element in building a successful speech recognition system.

Many different methods have been developed for feature extraction but most of them are based on cepstral analysis. Popular methods are LPC, cepstral coefficients, Mel frequency cepstral coefficients and perceptual linear prediction coefficients (PLP).

In this chapter, various methods of current feature extraction will be presented. In the next section the general properties of an ideal feature extraction process will be discussed. After that, cepstral analysis and Mel-frequency cepstral analysis techniques
which are widely used, are illustrated. LPC and line spectral frequencies are described subsequently. Another popular conventional feature extraction process, the PLP and Relative spectral processing are also described in this chapter. In the penultimate section of this chapter, as an essential step after feature extraction, feature normalisation and the employment of dynamic features are discussed.

3.2 Ideal feature extraction

As mentioned before, feature extraction has a great influence on the efficiency of a speech recognition system. It is the beginning stage of speech recognition after the preprocessing of speech. If this procedure does not operate properly or any small portion of necessary information is lost, there is no way to recover it in the later stages. An ideal feature extraction process should have the following properties:

1. Compression: It is the first and the most important property which a feature extraction process should possess. It is necessary that the raw speech signal is converted to a set of compact features which significantly simplifies the classifier task.

2. Removal of redundancy: A good feature extraction process should incorporate into the system information relevant to the recognition task and remove the redundant information.

3. Matching statistical characteristics of feature extraction process to those of the classifier. The inter-class variances of the vectors should be large while intra-class variances should be small for good recognition accuracy.

4. Improved compression and classifier efficiency is assisted by a decorrelation of features. A decorrelation of features leads to a diagonalised covariance matrix instead of full covariance matrix during the modelling of the feature coefficients. simultaneously.

As mentioned, these are the properties of an ideal feature extraction process. However, there is no method that includes all of them simultaneously. In general a feature
3.2. Ideal feature extraction

An extraction method can be modelled as follows:

\[ \mathbf{v} = \mathbf{T}[\mathbf{x}] \]  

(3.1)

where \( \mathbf{x} \) is the original speech signal and \( \mathbf{v} \) is the feature vector which is extracted from \( \mathbf{x} \) using the feature extraction method denoted by the process \( \mathbf{T}[.] \). This process may consist of several stages including the application of one or more orthogonal transforms and possibly a number of linear or non-linear operations. An example of one process, which has some of these ideal properties, is the Karhunen-Loeve expansion. This method involves just one operation in the process though some feature extraction methods such as cepstral analysis use to more than one operator (see section 3.3).

**Karhunen-Loeve expansion**

Suppose that \( \mathbf{x} \) is a random vector of dimension \( n \). The Karhunen-Loeve method states that the \( \mathbf{x} \) can be represented by a linear combination of \( m < n \) vectors from a set of orthonormal basis vectors \( \{ \mathbf{u}_i \} \), \( i = 1, 2, \ldots, m \) as follows:

\[ \hat{\mathbf{x}} = \sum_{i=1}^{m} y_i \mathbf{u}_i \]  

(3.2)

where \( y_i \) are the coefficients of expansion and can be computed as:

\[ y_i = \mathbf{u}_i^T \mathbf{x} \]  

(3.3)

These equations can be rewritten as:

\[ \hat{\mathbf{x}} = \mathbf{U}_m \mathbf{y} \]  

(3.4)

and

\[ \mathbf{y} = \mathbf{U}_m^T \mathbf{x} \]  

(3.5)

where \( \mathbf{y} \) is the \( m \)-dimensional vector whose components are the \( y_i \) coefficients and \( \mathbf{U}_m \) is the matrix of basis vectors (each column consists of one basis vector i.e \( \mathbf{U}_m = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m] \)). The problem now is to choose the basis \( \{ \mathbf{u}_i \} \) such that the mean square error \( e \) between vector \( \mathbf{x} \) and its approximation \( \hat{\mathbf{x}} \)

\[ e = E[(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T] \]  

(3.6)
Chapter 3. Overview of current feature extraction methods

is as small as possible. According to the orthonormal condition of basis vectors, it has
been shown in [86] that this can be achieved when the basis vectors are the eigenvectors
of \( E(xx^T) \) corresponding to the \( m \) largest eigenvalues. Thus the vector \( y \) which is
defined by Eq.3.5 contains most of the information about \( x \) and can be used as a
feature vector. The main properties of Karhunen-Loeve expansion are summarised as
follows:

1. Feature compression which reduces the computational complexity for pattern
classification.

2. Coefficients are uncorrelated: consistent with the orthogonality condition, the
features are uncorrelated [86] i.e:

\[
E[y_ky_j] = \begin{cases} 
\lambda_j & k = j \\
0 & \text{otherwise} 
\end{cases} 
\]  

(3.7)

where \( \lambda_j \) is the \( j^{th} \) eigenvalue.

The disadvantage of the Karhunen-Loeve, when applied to a non-stationary signal
as speech, is that the \( \{u_k\} \) may need to be up-dated on each frame. Under certain
circumstances it is shown that both the discrete cosine transform and Fourier transform
are close approximations to the Karhunen-Loeve and their substitution avoids this
difficulty [86, 27]. In deriving the Karhunen-Loeve feature, no special consideration is
given to the particular characteristics of a speech signal. To improve the overall feature
extraction performance, use of knowledge about speech production can be employed
with effect. One class of techniques which attempts this is the group of cepstral (or
homomorphic) processes which are described in the next section.

3.3 Cepstral analysis

Extensive research into feature extraction suggests that the cepstrum, and features
derived from the cepstrum, have been found to be more useful for speech recognition
compared to linear prediction. Cepstrum analysis was developed [12] as a technique
to examine data that was contained in echos of some arbitrary wave packet. Cepstral
3.3. Cepstral analysis

Cepstral analysis [14, 12], is a technique used for the separation of a convolved signal, such as the excitation and vocal tract impulse response of the speech. This is achieved by the transformation of the speech time domain samples into another domain called the quefrency domain in which the previously convolved components can be separated by linear techniques. The quefrency has dimensions of time. The method has found use in identifying both the pitch and spectral envelope of the magnitude spectrum of speech. Cepstrum techniques have been applied successfully to speech processing. The term cepstrum is essentially a coined word which includes the meaning of the inverse transform of the spectrum. The word quefrency also is formed from frequency. There are several types of cepstrum such as real cepstrum, complex cepsturm, difference cepstrum, root cepstrum. In the following sections, various types of cepstrum analysis will be discussed.

3.3.1 Real Cepstrum

The real cepstrum (or cepstral coefficients \( c(\tau) \)) is defined as the inverse Fourier transform of the logarithm of the magnitude of spectrum of a signal [71]. As we mentioned previously the speech signal is assumed to be the convolution of an excitation sequence \( e(t) \), with the impulse response of the vocal tract filter \( v(t) \).

\[
 s(t) = v(t) * e(t)
\]  

(3.8)

Figure 3.1: Block diagram for computing the Cepstrum
Chapter 3. Overview of current feature extraction methods

As shown in Fig. 3.1, the first step in this process is to perform a discrete Fourier transform in order to convert the convolution operator in the time domain to a multiplication operator in the frequency-domain, equation (3.9).

\[ S(w) = V(w) \cdot E(w) \]  

where \( S(w) \) is the Fourier transform of speech signal, \( s(t) \), after multiplication by a window function such as Hamming, \( V(w) \) is the Fourier transform of vocal tract impulse response \( v(t) \) and \( E(w) \) is the Fourier transform of excitation sequence \( e(t) \). The multiplication can then be converted to addition by taking logarithms of the magnitudes of both sides of (3.9)

\[ \log|S(w)| = \log|E(w)| + \log|V(w)| \]  

At a last step, the inverse Fourier transform is taken to produce the cepstral coefficients \( c(\tau) \):

\[ c(\tau) = \mathcal{F}^{-1}[\log|S(w)|] = \mathcal{F}^{-1}[\log|E(w)|] + \mathcal{F}^{-1}[\log|V(w)|] \]  

The excitation spectra and vocal tract spectra are now additive rather than convolved. Its lower quefrency part corresponds primarily to the shape of vocal tract, the glottal pulse and the lip radiation. Its higher quefrency part is primarily due to excitation. Since the shape of the vocal tract plays an important role in speech production, the lower quefrency part of the cepstrum is chosen as the feature vector for speech recognition.

Pitch estimation

Pitch is the fundamental frequency of a voiced speech waveform and is an important parameter in the analysis and synthesis of speech [90]. Pitch information can be extracted by using either temporal or frequency analysis. Based on frequency analysis methods, pitch can be determined from the periodic structure in the magnitude of the Fourier transform or in the cepstrum of a speech frame.

The process of separating the cepstral elements into fundamental frequency (pitch) and envelope is called liftering which is derived from filtering.
3.3. Cepstral analysis

The Cepstrum analysis offers a simple and accurate way for determining the pitch period of a frame of speech, by peak picking the high quefrency region of the cepstrum [70]. The process of liftering is shown in Fig.3.2.

![Figure 3.2: The process of liftering in cepstrum analysis](image)

**Spectral envelope**

As shown in Fig. (3.2), the spectral envelope may be obtained by firstly multiplying the cepstrum by a rectangular window of unit height and of a length long enough to contain all the low quefrency information required to form the spectral envelope. The exact length will depend upon the amount of detail required for the application. Finally, taking the Fourier transform of the liftered cepstrum yields $\log|V(w)|$, the log spectral envelope of the speech. In Fig.3.3 a frame of the speech signal and the analysis used to obtain the cepstrum and spectral envelope are shown.

Further, in Fig.3.4 more details about the cepstrum of a frame of voiced speech are shown. As we mentioned before, the real cepstrum provides two pieces of information, the impulse response of the vocal tract which is important for speech recognition and the pitch or excitation. As shown in this figure all of the information corresponding to the vocal tract is concentrated in the extremities of the quefrency axis and has a
Chapter 3. Overview of current feature extraction methods

3.3.2 Complex cepstrum

The real cepstrum provides a method for analysing the magnitude frequency of a signal after the phase has been discarded, therefore only part of the information corresponding to the magnitude is retained. The complex cepstrum provides an analysis method which retains all of the signal's information and therefore it should be a fully reversible process. The complex cepstrum of a speech signal $s(t)$ is defined in a similar manner to the real cepstrum as follows:

$$\hat{c}(\tau) = \mathcal{F}^{-1}\{\log \mathcal{F}[s(t)]\} = \frac{1}{2\pi} \int_{-\infty}^{\infty} \log|S(w)|e^{j\tau w} dw$$  \hspace{1cm} (3.12)

We know that the logarithm of the spectrum, $S(w)$ may be expressed as:

$$\log S(w) = \log |S(w)| + j\arg[S(w)]$$  \hspace{1cm} (3.13)
3.3. Cepstral analysis

By analysing equation (3.12) more carefully and considering the equation (3.13), it can be seen that the complex cepstrum is simply illustrated as the following:

\[ c(r) = \mathcal{F}^{-1}\{\log|S(w)|\} + \mathcal{F}^{-1}\{j(\arg[S(w)])\} \]  

(3.14)

The first term of Eq.(3.14) is the real cepstrum and the second term corresponds to the phase cepstrum. There is one difficulty with the complex cepstrum corresponding to the phase. The computation of the complex cepstrum requires a continuous function while the \( \arg[S(w)] = \tan^{-1}\{\text{Im}S(w)/\text{Re}S(w)\} \) is not a continuous function. A phase unwrapping algorithm is required [111] which is normally a difficult task, as phase unwrapping algorithms are seldom totally successful. In [9] an algorithm for the calculation of the complex cepstrum, which does not require phase unwrapping has been suggested. The issue of phase analysis and the inclusion of partial phase information in a feature matrix is discussed in chapter 6. There is some indication that phase may assist in the recognition task [83, 4].

3.3.3 Differential cepstrum

The differential cepstrum can be used to avoid of requirement of phase unwrapping in the complex cepstrum by computing the derivative of the logarithm and the phase of
Chapter 3. Overview of current feature extraction methods

\[ S(w) \text{ as follows [87]:} \]

\[ \frac{d[\log S(w)]}{dw} = \frac{1}{S(w)} \frac{d(S(w))}{dw} \]  

(3.15)

\[ \frac{d\arg[S(w)]}{dw} = \frac{d}{dw} \left\{ \tan^{-1} \left( \frac{S_i(w)}{S_r(w)} \right) \right\} \]  

(3.16)

\[ \arg'[S(w)] = \frac{S_r(w)S'_i(w) - S_i(w)S'_r(w)}{|S(w)|^2} \]  

(3.17)

where the prime denotes the \( d/dw \). The phase \( \arg[S(w)] \) can be defined in term of the integral of its derivative. Such a phase function, which is now a continuous function is commonly called the unwrapped phase [111]. Since the formulation of differential phase does not use of the arctangent function, the need for phase unwrapping is avoided.

### 3.3.4 Root cepstrum

A generalised class of process called Homomorphic De-convolution is shown in Figure 3.5. The purpose of such a system is to separate or de-convolve a signal, \( s(t) \) into its two

![Figure 3.5: Homomorphic deconvolution](image)

or more constituent parts. This may be achieved by a non-linear function, as shown previously with the real cepstrum, where a log-function was applied. A generalised form of non-linear function may be defined as:

\[ \tilde{S}(w) = |S(w)|^{\gamma} \]  

(3.18)

where \( \gamma \) can be any number, but of particular interest are the non-linear functions defined by:

\[ -1 \leq \gamma \leq +1 \]
The output of the homomorphic system when it is restricted to this range is referred to as the root cepstrum. The root cepstrum of a signal $s(t)$ may therefore be defined by:

$$\tilde{c}(\tau) = F^{-1}[\tilde{S}(\omega)]$$

(3.19)

The notion of root analysis in place of log was first proposed in [48], where the logarithmic Homomorphic De-convolution scheme is generalised to spectral root Deconvolution, by replacing the logarithmic and exponential with root and power operations. It has been proposed previously for speech recognition in a car-noise environment [3] and the robustness of the root cepstral analysis to noise has been demonstrated in [102].

**Pseudo cepstrum**

For the case when $\gamma = 1$ the root Cepstrum becomes the inverse Fourier Transform of the magnitude spectrum, rather than inverse Fourier transform of the log magnitude spectrum. This method has been called the Pseudo Cepstrum. It has been shown [42] that the discrete Pseudo cepstral coefficients are the Fourier series coefficients of a Fourier analysis of a magnitude spectrum, convolved with a set of impulses. In particular, the low order quefrency coefficients constitute the Fourier series coefficients of an estimated Fourier analysis of the magnitude spectral envelope of the speech signal. The plots in Figure 3.6 show how the spectral envelope of a frame of voiced speech can be separated from the excitation by this method.

### 3.4 Mel-Frequency cepstral coefficients

The mel-frequency cepstral coefficients (MFCCs) have been well reported in [21] and are commonly used in many speech recognition systems. Perhaps this method is the most popular of feature extraction techniques used for speech recognition today. MFCCs are based on the known variation of the human ear’s critical bandwidths with frequency. Mel-scale frequency is distributed linearly in the low frequency range but logarithmically in the high frequency range corresponding to the physiological characteristics of
the human ear. Figure (3.7) shows the block diagram of the Mel-frequency cepstral coefficients calculation. The details of Mel frequency analysis are expressed as follows.

### 3.4.1 Pre-emphasis

As a first step the speech signal is passed through a pre-emphasis filter which is a first order finite impulse response given by:

\[ H(z) = 1 - \alpha z^{-1} \]  

(3.20)

where \( 0.9 < \alpha < 1 \) and the most common value for \( \alpha \) is 0.95 [89]. In this case the output of the pre-emphasis network \( \tilde{s}(n) \) is related to the input of the network, \( s(n) \) by different equation:

\[ \tilde{s}(n) = s(n) - \alpha s(n - 1) \]  

(3.21)

The main aim of pre-emphasis is to spectrally flatten the speech signal and amplify important areas of the spectrum which are more sensitive for hearing. In fact equation (3.20) is a high pass filter which raises important frequencies and enhance the
3.4. Mel-Frequency cepstral coefficients

![Diagram showing the process of computing mel-frequency cepstrum coefficients]

Figure 3.7: Computing mel-frequency cepstrum coefficients
Chapter 3. Overview of current feature extraction methods

high frequency content. Figure (3.8) shows a frame of speech signal before and after preemphasis.

![Figure 3.8: Pre-emphasis](image)

3.4.2 Windowing and Spectral analysis

Spectral analysis is appropriate for stationary signals. While speech is non-stationary, it may be assumed stationary over a short time periods when the speech characteristics do not change. Therefore we apply windowing to the speech signal. Typically windows in signal processing may be rectangular, Hamming or Hanning [90]. The most commonly used window shape is Hamming because in this case side lobes of this window are lower than for other windows, which avoids leakage. Typically 10-25 ms windows are used in speech processing, since the changes in the vocal tract occur fairly slowly [89]. The window form is,

\[ W(n) = 0.54 + 0.46 \cos\left(\frac{2\pi n}{N - 1}\right) \]  

(3.22)
3.4. *Mel-Frequency cepstral coefficients*

After windowing the speech signal, a Fourier transform is applied to each frame and the short term power spectrum is computed as the following:

\[ P(\omega) = (\text{Re}[S(\omega)])^2 + (\text{Im}[S(\omega)])^2 \]  \hspace{1cm} (3.23)

where \( S(\omega) \) is the Fourier transform of speech signal.

3.4.3 Filter bank analysis

The linear frequency axis of the power spectrum obtained from the previous stage is then warped onto the Mel scale frequency axis which is defined by equation 3.24:

\[ \text{Mel}(f) = 1127 \log(1 + f/700) \]  \hspace{1cm} (3.24)

Figure (3.9) shows the Mel scale warping function. The warped magnitude spectrum is next passed through a bank of triangular-shaped filters as shown in Fig.3.10 which are the simulation of a critical band filter.

![Linear Frequency vs. Mel Frequency](image)

*Figure 3.9: The Mel Scale*

The filters are spaced linearly in the range 0 to 1000Hz. Above this the centre frequency is given by:

\[ f_{i+1} = 1.148f_i \]  \hspace{1cm} (3.25)

where the initial frequency \( f_1 \) is 1 kHz [64]. Each filter's magnitude frequency response is triangular in shape, and is maximum at centre frequency and decreases linearly to zero at the centre frequency of the two adjacent filter.
3.4.4 Logarithm

In the next step the logarithm function is applied to magnitude spectral values. In addition to a nonlinear response to frequency, the human ear responds to amplitude in a fashion which is approximately logarithmic, i.e. the ear is more sensitive to changes in amplitudes at low sound levels and less sensitive at high amplitude [109]. Thus application of logarithms to the magnitude of the spectral values in some way replicates this response. It also compresses the dynamic range of the power spectrum and assists in the efficient separation of vocal tract from excitation (see section 3.3).

3.4.5 Discrete cosine transform

The discrete cosine transform (DCT) is performed on the logarithm of the magnitudes of the filter bank outputs. This transform de-correlates the features, which leads to a diagonal covariance matrix instead of full covariance matrix during the modelling of the feature coefficients by a linear combination of Gaussian functions. Since the DCT gathers most of the information in the lower order coefficients, thus, by discarding the higher order coefficients, significant reduction in computational cost can be achieved. In conclusion, the Mel-Frequency Cepstral coefficients can be computed by using the
3.5. Linear prediction analysis

following equation:

\[ c_n = \sum_{i=1}^{L} \beta(\log(E(i)) \cos[n(i - 0.5)\pi/L]) \]

\[ \text{for } n = 1, 2, ..., P \]  

(3.26)

where \( E(i) \) is the energy output of the \( i^{th} \) filter, \( L \) is the number of filters in the desired band-width, \( \beta = \sqrt{2/L} \) and \( P \) is the total number of coefficients required. The effect of varying filter bank parameters are discussed in [20]. In [21], twenty such filters were used. The \( c_0 \) coefficient represents the average energy in the frame. It is often discarded to implement some form of amplitude normalisation. In order to combine dynamic properties of the speech signal, the first and second order of these coefficients (delta and delta-delta) may be used in many recognition systems to improve the recognition performance as will be discussed it in section (3.10).

3.5 Linear prediction analysis

Linear prediction analysis (LPC) has been considered as one of the most powerful techniques for speech analysis, coding and recognition [47] [52]. It is a time domain analysis compared to cepstral analysis and is different from previous methods. It can be used as a feature extraction method itself or used as a part of feature extraction method. Thus, this method is described briefly as follows. The basic idea is that given samples of speech, \( s[n] \), at time \( n \) can be approximated as a linear combination of the past samples [7]:

\[ s[n] \approx a_0 + a_1 s[n-1] + a_2 s[n-2] + ... + a_p s[n-p] = \sum_{k=1}^{p} a_k s[n-k] \]  

(3.27)

where \( a_0, a_1, ..., a_p \) are constant coefficients. In other words, speech can be modelled as the outputs of a linear time varying system excited by either periodic pulses for voiced or random noise for unvoiced speech as shown in Fig.3.11. The equation 3.27 can be written by including an excitation \( G u[n] \) as the following:

\[ s[n] = \sum_{k=1}^{p} a_k s[n-k] + G u[n] \]  

(3.28)
where $G$ is the gain and $u[n]$ the normalises excitation. By expressing the Eq.( 3.28) in the $z$-domain we will have:

$$S(z) = S(z) \sum_{k=1}^{p} a_k z^{-k} + GU(z)$$

(3.29)

The transfer function $H(z)$ which corresponds to the vocal tract transfer function can be written as:

$$H(z) = \frac{S(z)}{GU(z)} = \frac{1}{1 - \sum_{k=1}^{p} a_k z^{-k}} = \frac{1}{A(z)}$$

(3.30)

The prediction error between actual value of speech signal $s[n]$ and estimated speech signal $	ilde{s}[n]$ from equation (3.27) is expressed as:

$$e[n] = s[n] - \tilde{s}[n] = s[n] - \sum_{k=1}^{p} a_k s[n-k]$$

(3.31)

with error transfer function

$$A(z) = \frac{E(z)}{S(z)} = 1 - \sum_{k=1}^{p} a_k z^{-k}$$

(3.32)

The prediction error, $e[n]$ is equal to $Gu[n]$, the scaled excitation if the speech signal really does obey the model in Fig.3.11. The main goal is to find a set of predictor coefficients $\{a_k\}$ directly from the speech, so that $\{a_k\}$ minimises the square of the linear prediction error with respect to each of the parameters $a_k$. The two approaches
3.5. Linear prediction analysis

for obtaining the prediction coefficients are: autocorrelation and covariance methods and they are described in [89].

LPC analysis is based on the speech production model and the aim is to estimate a frame of speech by an all pole linear model. Since the system is assumed to be linear therefore it should be a reversible system. In other words the coefficients which are used as an approximation for any frame of a speech signal can be considered as a set of parameters used to recognise the corresponding frame. However, despite of wide use of LPC coefficients in different aspects of speech processing, the actual predictor coefficients are seldom used directly in recognition. The predictor coefficients are transformed to a set of parameters such as cepstral coefficients which are shown to be better than LPC coefficients for speech recognition[89]. Also we will see in section (3.7) that LPC analysis is used as a main part of Perceptual linear prediction representation which is used widely as a standard of feature extraction methods for speech recognition [35, 36].

3.5.1 Cepstral Coefficients derived from LPC

It is possible to derive cepstral coefficients from the linear prediction coefficients [29]. To establish the relationship, first we apply the logarithm to the LPC transfer function \( H(z) \) of an all pole model, given in equation (3.30):

\[
\log(H(z)) = \log\left[1 - \sum_{k=1}^{p} a_k z^{-k}\right]
\]  

(3.33)

By assuming that all the poles of \( H(z) \) lie inside the unit circle (necessary for stability) we can execute the Taylor expansion of \( \log H(z) \) which is guaranteed to converge:

\[
\log(H(z)) = c_0 + c_1 z^{-1} + c_2 z^{-2} + ... = \sum_{i=0}^{\infty} c_i z^{-i}
\]  

(3.34)

where \( \{c_i\} \) are the cepstral coefficients derived from LPC. By substituting for \( H(z) \) in equation (3.33) and apply the derivative with respect to \( z \) on both sides to get rid of the logarithm we will have:

\[
\sum_{i=0}^{\infty} i c_i z^{-i-1} = \frac{\sum_{k=1}^{p} k a_k z^{-k-1}}{1 - \sum_{k=1}^{p} a_k z^{-k}}
\]  

(3.35)
Multiply both sides of equation (3.35) by \(-z(1 - \sum_{i=0}^{\infty} c_t z^{-i})\) we obtain:

\[
\sum_{k=1}^{p} ka_k z^{-k} = \left[ \sum_{i=1}^{\infty} ic_i z^{-i} \right] \left( \sum_{k=0}^{p} a_k z^{-k} \right)
\]

letting \(a_0 = 1\) comparing the corresponding terms of the left and right side of equation (3.36), the coefficients can be determined as the follows [29]:

\[
c_0 = 0
\]

\[
c_1 = a_1
\]

\[
c_k = \sum_{i=1}^{k-1} (1 - i/k) a_i c_{k-i} + a_i \text{ for } 1 < k < p
\]

where \(c_i\) and \(a_i\) are the \(i_{th}\)-order cepstral and linear prediction coefficients respectively.

Note, that while there are a finite number of LPC coefficients, the number of cepstrum coefficients is infinite due to infinite number of \(c_i\) in equation (3.34) therefore for \(k > p\) we will have:

\[
c_k = \sum_{i=1}^{p} (1 - i/k) a_i c_{k-i} \text{ for } k > p
\]

Normally, in cepstral representation the number of coefficients which is used is around \(3p/2\) [89] where \(p\) is the number of LPC coefficients. This cepstrum is referred to as the LPC cepstrum, since it is derived through the LPC model. The original cepstrum is sometimes called the FFT cepstrum to distinguish it from the LPC cepstrum.

### 3.5.2 Comparison between LPC-cepstrum and FFT-cepstrum

Both LPC based cepstrum and FFT based cepstrum are dominant acoustic measurements of speech signals [85]. The LPC based approach performs spectral analysis with all pole modelling. It is faster than the FFT-cepstrum (direct method) and provides extremely accurate estimates of speech parameters and also offers almost the same results as FFT-cepstrum. On the other hand, FFT based cepstral coefficients represent a pole and zero model of speech. Moreover it is assumed in most applications that the speech signal is minimum phase so that the complex cepstrum can be estimated from only the Fourier magnitude spectrum [71]. The advantage of the FFT Cepstrum is its immunity to noise when warping the frequency to a non-uniform scale such as the mel.
3.6 Line spectral frequencies

Line spectral frequencies (LSF) also known as line spectrum pairs (LSP) were first introduced by Itakura [38]. In linear prediction analysis, as has been shown in section (3.5), a short stationary segment of speech signal is assumed to be represented by a time invariant linear all pole filter \( H(z) = \frac{1}{A(z)} \) where \( A(z) \) is called the inverse filter and expressed by:

\[
A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_p z^{-p}
\] (3.39)

Here \( p \) is the order of \( A(z) \) and \( \{a_i\} \) are LP coefficients. The inverse filter \( A(z) \) can be decomposed into two polynomials as the follows:

\[
P(z) = A(z) + z^{-(p+1)} A(z^{-1})
\] (3.40)

\[
Q(z) = A(z) - z^{-(p+1)} A(z^{-1})
\] (3.41)

The zeroes of \( P(z) \) and \( Q(z) \) can be expressed as \( e^{j\omega_i} \). These frequencies are called the line spectral frequencies and they uniquely characterise the LPC inverse filter \( A(z) \). They have the following properties:

1. All of the zeros of the \( P(z) \) and \( Q(z) \) are on the unit circle.

2. Zeros of \( P(z) \) and \( Q(z) \) are interlaced with each other.

3. Minimum phase property of all-pole filter is maintained if the properties (1) and (2) are preserved during the quantisation procedure.

4. It has been shown that LSP are related to the formant frequencies [82]. Because of these properties LSF representation is attractive and extensively used in the area of speech coding [107]. An efficient method to compute LSF from LPC coefficients has been proposed in [63].
3.6.1 Relationship between the LPC cepstrum and LSF

The relationship between the cepstrum and the LSF can be derived as follows. By multiplying equation (3.40) and equation (3.41) we will have:

\[ P(z)Q(z) = A^2[1 - R^2(z)] = (1 - z^{-2}) \prod_{i=1}^{p} (1 - e^{jw_i z^{-1}})(1 - e^{-jw_i z^{-1}}) \]  

(3.42)

where \( \{w_i\} \) are the line spectral frequencies and \( R(z) \) is defined as:

\[ R(z) = z^{-(p+1)}A(z^{-1})A(z) \]

(3.43)

By applying the logarithm to both sides of equation (3.42) we obtain:

\[ 2 \log A(z) + \log[1 - R^2(z)] = \log(1 - z^{-2}) + \sum_{i=1}^{p} \log(1 - e^{jw_i z^{-1}}) + \log(1 - e^{-jw_i z^{-1}}) \]

(3.44)

By taking the Fourier series expansions on both sides of equation (3.44) and considering 
\( \log A(z) = - \sum_{n=1}^{\infty} c_n e^{jwn} \) and doing some manipulation, the relationship between the LPC cepstrum, \( c_n \), and the LSF can be obtained as follows:

\[ c_n = 1/2n[1 + (-1)^n] + 1/n \sum_{i=1}^{p} \cos nw_i + R_n \quad \text{for} \quad n = 1, 2, ... \]  

(3.45)

where \( R_n \) is the inverse Fourier transform of \( \log(1 - R^2(z)) \). The first and second terms of the right side of equation (3.45) are expressed as \( \tilde{c}_n \) and called pseudo-cestral coefficients. Therefore \( c_n \) is defined as [43] [44]:

\[ \tilde{c}_n = 1/2n[1 + (-1)^n] + 1/n \sum_{i=1}^{p} \cos nw_i \quad \text{for} \quad n = 1, 2, ... \]  

(3.46)

3.6.2 LSF for speech recognition

LSF were proposed to be employed in speech compression and coding. In the context of speech coding, it shows better quantisation properties than the LP parametric representation. However, researchers have started to investigate the flexibility of using LSF for speech or speaker recognition [106] [13]. It is possible to derive a number of parameters from the LSF analysis of speech, containing suitable information about the speech signal for speech recognition. Different distance measures, based on LSF representation
for speech recognition, have been considered and it has been shown that LSF representation has better recognition performance than formant representation [80]. In [81] the LSF representation is used as the parametric representation for speech recognition and its performance is compared with that of the cepstral coefficients representation in both speaker and speaker independent modes for the hidden Markov model based isolated word recognition speech. A feature set form suitable for speech recognition obtained from quantised LSF parameter in code excited linear prediction coder (CELP) has been suggested in [18].

3.7 Perceptual linear prediction representation

Another popular speech feature representation is known as Perceptual Linear Prediction representation (PLP). PLP analysis is a variation of original LPC analysis and was first introduced by Hermansky [35]. It uses the concepts from the Psycho-acoustics of human hearing and yields lower dimension representation of speech, which are found to be useful for a speaker independent recognition system. Three properties of the human auditory system are implemented in PLP; the nonlinear frequency response of the human ear; the critical bands in the cochlea; and the non-linear amplitude response. Figure (3.15) shows the blocks for performing the PLP analysis. The details of the steps involved in PLP technique are illustrated in the following.

3.7.1 Spectral analysis

The speech signal is divided into frames using a Hamming window. The Fourier transform is applied to each frame and the short term spectrum is computed as follows:

\[ P(\omega) = (Re[S(\omega)])^2 + (Im[S(\omega)])^2 \]  

(3.47)

where \( S(\omega) \) is the Fourier transform of speech signal.

3.7.2 Critical band analysis

The linear frequency axis is transformed to the Bark scale frequency which is an auditory-based warping of the frequency axis derived from the frequency of human
Chapter 3. Overview of current feature extraction methods

Figure 3.12: Perceptual linear prediction representation of speech signal
3.7. Perceptual linear prediction representation

hearing. The Bark scale frequency is defined as follows:

\[ \Omega(\omega) = 6 \ln \left\{ \frac{\omega}{1200\pi} + \left[ \left( \frac{\omega}{1200\pi} \right)^2 + 1 \right]^{1/2} \right\} \]  

(3.48)

Figure (3.13) shows the Bark-scale warping function.

![Bark-scale warping function](image)

Figure 3.13: Bark-scale warping function

The resulting of warped power spectrum is passed through a set of simulated critical band filters which are defined by the equation (3.49).

\[ \Phi(\Omega) = \begin{cases} 
0 & \text{for } \Omega < -1.3 \\
10^{2.5(\Omega-0.5)} & \text{for } -1.3 < \Omega < -0.5 \\
1 & \text{for } -0.5 < \Omega < 0.5 \\
10^{-1.0(\Omega-0.5)} & \text{for } 0.5 < \Omega < 2.5 \\
0 & \text{for } \Omega > 2.5 
\end{cases} \]  

(3.49)

This simulates the frequency resolution of the ear which is approximately constant on the Bark scale. This critical band mask curve is an approximation to the asymmetric mask curves of Schroeder[35]. As with the MFCC analysis, more filters are allocated to the lower frequencies, where hearing is more selective. Fig.3.14 shows critical band basis functions.
3.7.3 Equal loudness pre-emphasis

In order to compensate for the unequal sensitivity of human hearing across different frequencies, the critical bands amplitudes are scaled according to an equal loudness pre-emphasis function such as:

$$H_E(\omega) = \frac{(\omega^2 + 56.8 \times 10^6)\omega^4}{(\omega^2 + 6.3 \times 10^6)(\omega^2 + 0.38 \times 10^9)}$$  \hspace{1cm} (3.50)

where $\omega$ represents the angular frequency in radians.

3.7.4 Intensity-Loudness power law

This processing stage models the non linear relation between the sound intensity and its perceived loudness. This function is implemented by cubic root compression of critical band energies as follows:

$$L(\Omega) = E(\Omega)^{1/3}$$  \hspace{1cm} (3.51)

This operation provides low order all pole modelling due to reduction of spectral amplitude variation of the critical band spectrum (In the next stage). It is clear that low order pole modelling will reduce the computational cost.
3.7.5 Autoregressive modelling

In this stage the result is modelled by an all pole filter. Once the spectrum is obtained it is converted back into the time domain and an auto correlative all pole LP analysis is performed to obtain the PLP coefficients. The filter coefficients \(\{a_1, a_2, ..., a_p\}\) form a prediction filter \(A(z)\)

\[
A(z) = 1 - \sum_{k=1}^{p} a_k z^{-k}
\]  

(3.52)

The auto correlative function can be obtained from the IFT of the power spectrum. Durbin's recursion algorithm can then be applied to obtain the prediction coefficients which minimise the error.

3.7.6 Post-processing

The coefficients which are derived in the last stage can be transformed into the other types of parameters for further analysis such as cepstral analysis and used as a feature set for speech recognition or directly can be used in the pattern recogniser of speech recognition system.

3.8 Relative spectral processing of speech

The word RAST stands for RelAtive SpecTrAl processing of speech which was proposed by Hermansky as an operation to reduce both communication channel effects and noise distortion [36]. RASTA is a technique that applies a 4th order IIR filter to the energy in each frequency sub-band of the critical band filter in order to smooth over short-term noise variation and to remove any constant offsets. Several improvements have been proposed, such as J-RASTA, phase corrected RASTA and automatic computation of the filters [46] [22]. This filtering method is used in conjunction with the PLP method using some additional steps.
3.8.1 RASTA-PLP technique

This technique is an improvement of the traditional PLP method and consists of a special filtering of the different frequency channels of a PLP analyser as described in section (3.7). Fig.3.15 shows the block diagram of RASTA-PLP speech processing. The steps involved in RASTA-PLP are illustrated as the following [36]:

1. Compute the critical-band power spectrum as in PLP analysis.
2. Transform spectral amplitude through a compressing static nonlinear transforma-
tion. After computing the critical band power spectrum in PLP, it is transformed by a compressing nonlinear function, such as the logarithm function.

3. Filter the time trajectory of each transformed spectral component. In this step each transformed spectral amplitude is then filtered by the RASTA filter described in [36]. The goal in this filter is to suppress constant factors and slowly varying component by introducing sharp spectral zeros at the zero frequency.

4. Transform the filtered speech through expanding static nonlinear transformations. In this step the result is expanded by using a nonlinear transformation such as the inverse function of the one used while compressing. (step 2)

5. Multiply by the equal loudness curve and raise the power 0.33 to simulate the power law of hearing.

6. Compute an all-pole model of the resulting spectrum as in the PLP technique.

RASTA bandpass filter may not effectively remove uncorrelated additive components in the signal because they become signal dependent after the logarithm operation. A signal-dependent positive constant, $J$, may be used to transform the power spectrum to a linear-log like domain (in step 2 in the process of computing RASTA-PLP) using the following equation [66]:

$$y = \ln(1 + Jx) \quad (3.53)$$

The amplitude-warping transform is linear-like for $J \ll 1$ and logarithmic-like for $J \gg 1$. The exact inverse of (3.53) is

$$x = \frac{e^y - 1}{J} \quad (3.54)$$

and is not guaranteed to be positive for all $y$. To avoid negative values, the expanding function in equation (3.55) uses an approximate inverse transform in expanding the nonlinearity, in the step 4 of RASTA processing instead of its exact inverse.

$$x = \frac{e^y}{J} \quad (3.55)$$

The optimal value for $J$ is dependent on the instantaneous noise. There is a distinct optimal value of $J$ for each particular noise level. More detail can be found in [46, 37, 66].
3.9 Feature Normalisation

Following the computation of static features, most speech recognition systems employ one or more forms of feature normalisation to the feature vector. Feature normalisation techniques are used for the removal of unknown linear filtering effects without removing useful speech information. Normalisation can be done either in the spectral domain or the cepstral domain.

3.9.1 Cepstral mean normalisation

Cepstral mean normalisation (CMN) which also sometimes is referred to as the cepstral mean subtraction (CMS) [104], is a technique to high-pass filter cepstral coefficients and to remove the convolution noise from a speech signal by subtracting the mean of cepstral feature vectors from each vector. Because convolution in the time domain is equivalent to addition in the logarithmic spectral domain, subtracting an estimated noise log spectrum from the log spectral or cepstral features can remove the convolution noise. Given a speech signal \( s(n) \), we compute its cepstrum through short-time analysis, resulting in \( N \) cepstral coefficients \( C = [c_1, c_2, ..., c_N] \). Its sample mean \( m_c \) is given by:

\[
m_c = \frac{1}{N} \sum_{i=1}^{N} c_i
\]

The CMN coefficients can be found by following:

\[
\bar{c}_i = c_i - m_c
\]

where \( c_i, \bar{c}_i \) are the cepstrum and normalised cepstrum coefficients respectively. CMN can be viewed as performing a type of high pass filtering of the temporal characteristics of the signal spectrum, to remove the constant component. As we mentioned, this method can be applied to parameters other than cepstral coefficients such as the spectral domain.

3.10 Dynamic features

The acoustic features described up to now can be called static features since they aim to represent the power spectrum of a frame of speech signal which is assumed to represent
3.10. Dynamic features

a stationary signal. It has become common to combine these features with dynamic features that account for a longer interval for time evolution of the power spectrum. These dynamic features are often referred to as time derivatives of corresponding static features or delta features [30, 92]. In the following section, we will briefly explain the delta and delta-delta cepstrum which are commonly used in speech recognition systems.

3.10.1 Delta cepstrum

ASR performance is often greatly improved by using the delta cepstrum. One way of computing the delta features is by simple differencing between the feature values for two frames either side of current frame:

\[ d_n = c_{n+w} - c_{n-w} \]  

(3.58)

where \( d_n \) is a delta coefficient at frame \( n \), and \( w \) represents the number of frames to offset either side of the current frame [33]. Typically \( w \) is set to a value of 1 or 2 and \( c_{n+w} \) and \( c_{n-w} \) are static parameters before and after to the current frame coefficient \( c_n \) [100].

Though time-difference features have been used successfully in many systems, they are sensitive to random fluctuations in the original static features and therefore may be mixed with noise. A more robust measure of local change is obtained when linear regression is applied over a sequence of frames as follows [54] :

\[ d_n = \frac{\sum_{w=1}^{\delta w} w (c_{n+w} - c_{n-w})}{2 \sum_{w=1}^{\delta w} w^2} \]  

(3.59)

where the \( d_n, c_{n+w}, \) and \( c_{n-w} \) have the same definition as in equation (3.58) and \( \delta w \) is the delta window size. While the regression technique requires past and future speech parameter values, suitable modification can be performed at the beginning and at the end of the data stream. The Delta cepstrum is usually included into feature vectors so that dynamic features can be taken into account. The Delta cepstrum indicates the variation of cepstral coefficients from frame to frame. It is a general way of expressing the variation of local features.
3.10.2 Delta-Delta cepstrum

The delta features, described in the previous section, are first-order time derivatives. They can be used in turn to calculate second-order time derivatives or delta-deltas which are also called acceleration coefficients. By including the first time derivative features a large gain in recognition performance is obtained, and adding second-order derivatives, which capture changes in the first-order dynamics gives additional improvement [30]. Most current recognition systems use first-order derivative features and often apply them to basic feature sets of MFCC (or other types of features) and some also include second-order derivatives to get additional improvement. Usually, between 8 and 16 cepstral coefficients, together with their first and second order time derivatives are considered as the speech extended feature vectors which are supplied to a modern speech recognition system.

3.11 Summary and discussion

This chapter reviewed different approaches used as the basis for most feature extraction methods in current speech recognition systems. Most of these methods involve the cepstral analysis as the main part of the feature extraction process. Cepstral analysis attempts to compress as much information into the lower order quefrency indexes by applying a non linear operator to the spectrum such as a log function. Several cepstral analysis techniques for feature extraction have been discussed. Since the real cepstrum presents a method based on magnitude spectrum of a speech signal, some information corresponding to the phase is lost. The complex cepstrum offers an analysis method in which all of the speech's information is retained, but, it needs a phase unwrapping algorithm which is normally a difficult task. To overcome the phase problem the differential cepstrum has been suggested.

LPC is another dominant type of speech representation which is used widely in speech processing. Both linear prediction analysis and cepstral analysis are based on the speech production model whereas the physiology of human ear provides the basis for filter bank analysis which is used in Mel cepstral coefficients and PLP. In other words
the knowledge gained from studying human auditory processing has resulted in these techniques. Comparing cepstral methods to LPC, it should be noted that all cepstral techniques can represent the speech signal based on poles and zeros while the LPC are only based on an all pole model. Thus, it is clear that a frame of speech signal can be better represented with the cepstrum which contains both pole and zeros information. Although cepstral coefficients are widely used in speech recognition, the LPC coefficients are seldom used directly in speech recognition applications. However, it can be applied as a main part of Perceptual Linear Prediction as described in section (3.7).

Among different types of feature extraction methods, which are mentioned in this chapter, MFCC and PLP are the most popular. Both methods employ an auditory-based warping of the frequency axis, derived from the frequency sensitivity of human hearing. MFCC are based an a uniform spacing along the mel-scale whereas PLP uses the Bark scale. However there are significant differences in the process of computing the PLP coefficients and MFCC coefficients. The main difference between PLP and MFCC is related to the output cepstral coefficients. The PLP model uses an all pole model to smooth the modified power spectrum. In contrast, Mel scale cepstral analysis uses cepstral smoothing to smooth the modified power spectrum. PLP provides a more robust representation than MFCC yet in spite of this, MFCC are historically more popular probably because MFCC is simpler to implement. More details about the comparison between PLP and MFCC can be found in [65].

Finally, the feature normalisation and dynamic features are discussed briefly in this chapter.

While some feature extraction methods involve extracting an instantaneous or static feature from a speech signal it is necessary to apply further step to obtain more information from the speech. Delta and Delta-Delta are a measure of rate of change of a feature and are used to establish the time dependent changes between two frames of features.
Chapter 3. Overview of current feature extraction methods
Chapter 4

Two-dimensional cepstrum analysis

4.1 Introduction

Generally, speech can be represented in one of two dimensions: time scale or frequency scale. Therefore, the speech signal features consist either of the features that are related to the time domain of the signal or features that are related to signal’s frequency domain. Examples of features that belong to the first category are short time average zero-crossing rate, short time energy and linear prediction coefficients, while spectral analysis, cepstral analysis, Mel-frequency cepstral analysis are some of the more important examples of methods which have been used in the second category [90, 101]. Most current feature extraction techniques involve a one-dimensional feature, usually a one dimensional cepstrum, to represent the speech signal. In the previous chapter the different types of feature extraction method for speech recognition, all of them based on one dimensional features, were discussed.

The Two-Dimensional Cepstrum (TDC) which has been introduced recently [5], is an alternative approach for time frequency representation of speech signals. It is a special case which can represent both the instantaneous and transitional information of a speech wave form.
Chapter 4. Two-dimensional cepstrum analysis

In this chapter, the different types of two dimensional cepstrum will be discussed initially. Next, as a novel method of feature extraction, the Two Dimensional Root Cepstrum (TDRC) and its properties is introduced, followed by a discussion of the complex TDRC.

4.2 Why Two-dimensional features?

The research, presented in this chapter, of time frequency representation has been of interest because detailed information, from non-stationary signals such as speech, in both the time and frequency domain, potentially can improve the recognition process. As mentioned in chapter 3, conventional features are one dimensional either time or frequency and are widely used in speech recognition. To include the speech transitional dynamics, it is necessary to apply further steps to obtain more information from the speech by using delta or delta delta features as discussed in the previous chapter. It has been shown that speech characteristics may be better represented by an acoustic image, a two dimensional feature representation with time along one axis and frequency along the other [61, 59].

Spectrograms are the most familiar form of acoustic image, though they are seldom used for speech recognition, but other forms such as the two-dimensional cepstrum have more useful characteristics in speech recognition. By using the two-dimensional features such as the TDC, the dynamic features and static features are represented simultaneously. The TDC represents the features of the speech signal in a matrix form. Analysis results show that the coefficients located at the lower index portion of the TDC matrix seem to be more significant than others. Hence to represent an utterance only some TDC coefficients need to be selected to form a feature matrix [77].

In addition, the TDC has the capability of separation of vocal tract information from excitation information unlike the spectrogram. Of course the MFCC, conventionally used in speech recognition, also has the same property, but do so on a frame-by-frame basis. The key advantage of the TDC is that this information is represented over the entire utterance and also has the property of data compression which makes the classification process more reliable.
4.3 Approaches to two dimensional cepstrum analysis

There are 3 alternative definitions for the TDC mentioned in the literature. These three definitions are described below.

4.3.1 Cepstrum of two dimensional signal

The first is the cepstrum of two dimensional signals mostly used for image processing. The cepstrum for the two dimensional signal is defined in the same way as for the definition for a one dimensional signal. It can be defined for a separable transform $T$ such as Fourier transform and a two dimensional signal, $s(m,n)$, as follows:

$$c(u,v) = T^{-1}\{\log|T(s(m,n))|\} \quad (4.1)$$

The absolute value has been used, since $T(s(m,n))$ is in general a complex valued and non-positive and the log function in the complex domain is not single valued. If the absolute value in (4.1) is deleted; the result is called the two dimensional complex cepstrum. The most significant property of the two dimensional cepstrum as with the one dimensional cepstrum, is the property of separability of two convolved signal. This kind is suitable for two dimensional functions such as images. More details can be found in [99].

Differential two-dimensional cepstrum

The main disadvantage of the complex cepstrum, as mentioned in section (3.3.2) is the necessity for a complicated phase unwrapping algorithm [111]. Polydoros and Fam [87] introduced the one dimensional differential cepstrum to overcome this drawback. In the same way the two dimensional differential cepstrum has been introduced in [93] as follows. Let again $s(m,n)$ be a two dimensional signal and its two dimensional $z$ transform is defined by:

$$S(z_1,z_2) = \sum_m \sum_n s(m,n)z_1^{-m}z_2^{-n} \quad (4.2)$$
Chapter 4. Two-dimensional cepstrum analysis

By taking the partial derivatives from the logarithm of $S(z_1, z_2)$ with respect to $z_1$ and $z_2$ we will have:

$$\hat{S}_d(z_1, z_2) = \frac{\partial S(z_1, z_2)/\partial z_1 + \partial S(z_1, z_2)/\partial z_2}{S(z_1, z_2)} \quad (4.3)$$

Thus the two dimensional differential cepstrum is defined as the inverse two-D transform of $\hat{S}_d$ as follows:

$$c(m, n) = (1/2\pi f)^2 \int_{C_1} \int_{C_2} \hat{S}_d(z_1, z_2) z_1^{m-1} z_2^{n-1} dz_1 dz_2 \quad (4.4)$$

This definition of a two dimensional cepstrum includes both spectral magnitude and phase information while avoiding the need for phase unwrapping.

4.3.2 Cepstrum time matrices

Second is the two dimensional cosine cepstrum usually called the cepstral time matrix (CTM) [114]. The two-dimensional cepstrum time matrix (CTM) can be obtained by applying the two-dimensional discrete cosine transform (DCT) to a log spectral time matrix [114, 73]. At the first step, the speech signal is segmented into the $M$ overlapping frames with $K$ samples in each frame and each frame is transformed to the spectral domain by applying a discrete Fourier transform. These $K$ spectral samples are grouped into $N$ overlapping, mel scaled, triangular frequency bands, and the samples within each band are averaged to form power spectral features. Then, these power spectral features are converted to log-power by applying the log function. The result is an $N \times M$ log spectral-time feature matrix, $S_i(f, t)$ where $f$ denotes the frequency domain and $t$ denotes the time domain. The log spectral-time matrix is transformed via a 2-dimensional DCT to the cepstral-time matrix $c(u, v)$

$$c(u, v) = \sum_{t=0}^{M-1} \sum_{f=0}^{N-1} S_i(f, t) \cos\left[\frac{\pi}{2N} (2f + 1)\right] \cos\left[\frac{\pi}{2M} (2t + 1)\right]$$

$$0 \leq u \leq N - 1, \quad 0 \leq v \leq M - 1 \quad (4.5)$$

Since the two dimensional DCT can be decomposed in two one dimensional DCTs, the CTM computation can be performed by applying two one dimensional DCTs where one DCT operates on the log spectral vectors (row of matrix) and the other DCT
4.3. Approaches to two dimensional cepstrum analysis

operates on the time variation of the cepstral coefficients (column). Following the DCT operation a sub-matrix $N' \times M'$ is chosen to represent the speech signal which contains a set of coefficients most useful for speech recognition [114].

The CTM has been reported in the literature. A comparative evaluation of the use of noise-compensation methods for HMM using cepstral vectors and HMMs using cepstral-time matrices has been reported in [112, 72]. The results demonstrate that the noise-compensated CTM generally outperforms noise-compensated cepstral vectors. Further in [113] the use of CTM features with spectral subtraction and state-based time varying Wiener filter is investigated. It is shown that the CTM provides an effective framework for development of noise compensation systems. Also, as illustrated in [74], the CTM can be made robust to channel distortion.

4.3.3 Two dimensional cepstrum

The third method is the two dimensional cepstrum first reported in [5] and is developed in this thesis. The TDC is defined as the two-dimensional inverse Fourier transform of the log spectrum of a speech signal [116, 77]

$$c(u, v) = \frac{1}{NM} \sum_{m=0}^{N-1} \sum_{k=0}^{M-1} S_i(m, k)e^{j2\pi kv/N}e^{j2\pi mu/M}$$

$$0 \leq u \leq N - 1, 0 \leq v \leq M - 1$$

(4.6)

where the $S_i(m, k)$ is given by:

$$S_i(m, k) = \log{|S(m, k)|}$$

(4.7)

and $S(m, k)$ is the Fourier transform of $s(m, n)$, with $n$ being the $n^{th}$ sample in frame $m$ of the speech signal, $N$ is the number of samples in a frame, and $M$ is the number of frames used for computing the TDC matrix. As the $c(u, v)$ is a complex value the TDC matrix coefficients are defined as follows:

$$c(i, j) = |c(u, v)|$$

(4.8)

where the $i, j$ indicate the row number and column number in the TDC matrix. The axis $v$ is called quefrency and has time dimension. The axis $u$ is called time frequency.
and has frequency dimension. Figures (4.1) and (4.2) show the 3-D representation of the $32 \times 16$ sub-matrix which has taken from the TDC matrix for the spoken digits One to Eight. As shown in these figures, significant values are concentrated in the low quefrency and low frequency region.

Figure 4.1: 3-D representation of TDC matrix for digits: one, two, three and four

The TDC has been reported in the literature as follows. Pai and Wang [76] [75][79] have suggested the use of a TDC based Bayesian classifier for the recognition of vowels and isolated Mandarin utterances in a speaker dependent manner.

In [78] a method based on the TDC distance measure and vector quantisation is suggested for classifying TDC patterns where the distance measure between cepstral features has been used. This method has been demonstrated by applying it to the recognition of Mandarin digits. The results show that the proposed method is a very promising one for the speech recognition of syllabic languages such as Mandarin Chinese.
4.3. Approaches to two dimensional cepstrum analysis

Figure 4.2: 3-D representation of TDC matrix for digits: five, six, seven and eight
Lin and Nein and Hwu [49] have used Mel TDC for noisy speech recognition by applying the Genetic Algorithms to find the robust coefficients in the Mel TDC matrix. They found that the GA-based Mel TDC has better recognition results than the original TDC approach in noisy environments. In [45] a speaker recognition model using a TDC Mel-cepstrum and predictive neural network has been proposed. It has been shown in this paper that the TDC Mel-cepstrum is very effective for speaker recognition. In [68] the performance of the two-dimensional cepstrum has been compared with delta feature. It has been shown that the two-dimensional cepstrum can significantly outperform conventional delta delta features, especially in mismatches (there is mismatch when the conditions of training are different from those of testing) environment. Moreover in [67] the success of the two-dimensional cepstrum in capturing the dynamics of speech sounds has been demonstrated. Using this feature set, up to 92% recognition accuracy has been achieved with the highly confusable E set (b,c,d,e,g,p,t,v,z). The application of TDC as a static and dynamic feature for recognition of English digits in a speaker independent manner have been suggested in [55]. In [60] the one-dimensional cepstrum features has been compared to the TDC features using different kinds of distance measure. We have shown in this paper that the recognition accuracy of TDC features are much better than that of the one dimensional features for all distance classifier.

4.3.4 Schemes for computing the TDC

There are three schemes for computing the TDC matrix. In the first method the utterance is divided into consecutive frames and several frames are then grouped as a block. The whole utterance is represented by several blocks. In [5], each frame is 256 samples (25.6 ms) long with a frame period of 64 samples (6.4 ms) i.e a frame overlap of 192 samples (19.2ms). Eight consecutive frames form each block with a block period of four frames. The TDC coefficients are evaluated for each block. Dynamic programming (DP) is used to match a time sequence of the TDC. This method is complicated due to the use of the DP and it requires more memory for storing the reference patterns.

In the second, a fixed number of frames, of pre-determined size, are selected from the entire utterance by linear sampling and selected regardless of the length of the utterance. This means that the fixed number of frames are selected at uniform intervals
4.3. Approaches to two dimensional cepstrum analysis

across the utterance e.g if the chosen number of frames is 128 (with each frame being 256 samples in length ) and the entire utterance is 5000 samples long, then each frame starts at intervals of 39 samples will an overlap of 217 samples. Depending on the length of the utterance some data may be ignored for long utterances while may be included in more than one frame where utterances are short. A single block of data is used for each utterance and the TDC is computed for this block. The memory required for storing the reference patterns is small because utterance features are represented in a single word matrix [5].

In the third method [77], which we adopt here, the TDC matrix is calculated using the entire utterance without any frame deletion. Since the entire utterance is used, all the characteristics are properly preserved. Furthermore, in this method, each utterance is represented by one feature vector which is extracted from the TDC matrix rather than a string of feature vectors.

Although the TDC incorporates both time and frequency information, it should be conceded that in the TDC method described here, some of the information is lost in obtaining the 2-D cepstrum, in particular there may be some loss of time ordering, so that a pair of utterances such as 'six' and 'kiss' may be confused. This was not investigated.

Another drawback may be that this method does not readily lend itself to continuous speech recognition. Since it would be difficult to identify suitable word boundaries in the continuous speech, one approach would be that reported in [64] where boundaries located in areas of spectral stability could be readily identified and analysis performed on syllable units.

4.3.5 Comparison between TDC and CTM

Although the CTM can be considered as a kind of TDC, there are two main differences between them.

1. Firstly the TDC is applied to the two-dimensional FT while the CTM is applied to the two dimensional DCT.
2. Secondly, the CTM uses a mel frequency scale while the TDC uses only a linear scale.

If we assume a linear frequency scale for both of TDC and CTM then we will see that Eq.(4.5) is closely related to Eq.(4.6). By considering $e^{jx} = \cos(x) + jsin(x)$ and expanding the eq(4.6) we have:

$$\hat{s}(u,v) = \frac{1}{NM} \sum_{m=0}^{M-1} \sum_{k=0}^{N-1} S_t(m,k)[\cos(x) + jsin(x)][\cos(y) + jsin(y)]$$

where $x = \frac{2\pi kv}{N}$ and $y = \frac{2\pi mu}{M}$. After some manipulation we have:

$$\hat{s}(u,v) = \frac{1}{NM} \sum_{m=0}^{M-1} \sum_{k=0}^{N-1} S_t(m,k)((\cos x \cos y - \sin x \sin y) + j(\cos x \sin y - \cos y \sin x))$$

(4.10)

The real part of Eq.(4.10) is:

$$\hat{r}(u,v) = \frac{1}{NM} \left\{ \sum_{m=0}^{M-1} \sum_{k=0}^{N-1} S_t(m,k)\cos x \cos y - \sum_{m=0}^{M-1} \sum_{k=0}^{N-1} S_t(m,k)\sin x \sin y \right\}$$

(4.11)

By comparing the first part of the above equation and Eq.(4.5), assuming $f = m, t = k$ it is clear that these two are almost the same.

### 4.4 Regions of TDC

The lower components on the $u$-axis of the TDC matrix represents the spectral envelope while the higher components illustrate the pitch and excitation signals. This is the same as in the conventional one dimensional cepstrum. The lower components on the $v$ axis express the long time variation, while the higher components represent the short time variation of cepstral variables [5]. The coefficient $c(0,0)$ is relate to the energy of the log spectrum while the $c(0,v)$, is related to the energy of each frame. To avoid the influence of speech energy these coefficients are not used in speech recognition. The average of quefrency coefficients over the time index are represented by $c(u,0)$. These coefficients are also excluded from the feature sub matrix due to having large variance. In general the TDC plane is divided to 9 regions as shown in Fig.4.3. These regions are illustrated as follows [5]:

...
4.4. Regions of TDC

$A_1$: $u = 0, \quad v = 0$ Averaged value of the log spectrum in a matrix

$A_2$: $u = 0, \quad 0 \leq v \leq m$ Global time variation of the averaged log spectrum

$A_3$: $u = 0, \quad m \leq v \leq M/2$ Local time variation of the averaged log spectrum

$B_1$: $0 \leq u \leq n, \quad v = 0$ Spectral envelope

$B_2$: $0 \leq u \leq n, \quad 0 \leq v \leq m$ Global time variation of the spectral envelope

$B_3$: $0 \leq u \leq n, \quad m \leq v \leq M/2$ Local time variation of the spectral envelope

$C_1$: $n \leq u \leq N/2, \quad v = 0$ Spectral fine structure

$C_2$: $n \leq u \leq N/2, \quad 0 \leq v \leq m$ Global time variation of the spectral fine structure

$C_3$: $n \leq u \leq N/2, \quad m \leq v \leq M/2$ Local time variation of the spectral fine structure

It is noted that the higher order of TDC regions i.e $A_3, B_3, C_3, C_2, C_1$ have no relevance for recognition and can be ignored in the feature matrix. Thus, the TDC dimensions have an important effect on recognition accuracy and computational complexity which are discussed in chapter five.

Figure 4.3: Regions of two-dimensional cepstrum
4.5 Two-dimensional root cepstrum analysis

The TDC as defined in the section 4.3.1 is evaluated using a log function applied to the magnitude spectrum. There is one weakness in this method arising from the use of the log function. Since, as \( x \) tends to zero, \( \log(x) \) tends to minus infinity, the function is very sensitive to small values of \( x \). In the cepstrum this means that there is most sensitivity to those parts with lower spectral power, i.e. to those parts where the SNR is normally worse. One well known technique for dealing with this problem is to replace \( \log(x) \) function with root function, \( x^{\gamma} \) where \( -1 < \gamma < +1 \) [48]. Therefore we introduce the two dimensional root cepstrum [17] as a novel feature extraction technique. It has the advantage of an adjustable \( \gamma \) parameter which can be used to enhance the performance. Further, it has the benefit of reducing the dimensionality of the feature matrix when compared to more conventional techniques [103] (See Fig 4.8). Furthermore it is robust to noise and is simple to compute.

The definition of TDRC is introduced in the next subsection followed by a summary of its properties.

4.5.1 Definition of TDRC

The TDRC can be defined as the two-dimensional inverse Fourier transform of the root of the magnitude spectrum of a speech signal [17]. It can be obtained by the following stages as shown in Fig.4.9.

Stage (1) Preprocessing the speech including: Low pass filter, pre-emphasis.

Stage (2) Block the speech signal to frames.

Stage (3) Apply the Hamming window

Stage (4) Take the discrete Fourier transform of the samples of each frame:

\[
S(m, k) = \sum_{n=0}^{N-1} s(m, n)e^{-j2\pi nk/N} \quad 0 \leq k \leq N - 1 , 0 \leq m \leq M - 1 \quad (4.12)
\]
4.5. Two-dimensional root cepstrum analysis

Figure 4.4: Two dimensional root cepstrum analysis of speech signal
Chapter 4. Two-dimensional cepstrum analysis

In equation (4.12) \( s(m,n) \) is the \( n^{th} \) sample of speech signal in frame \( m \), \( N \) indicates the number of sample data in a frame, and \( M \) represent the number of frames which are used to compute the TDRC matrix.

Stage (5) Compute the magnitude of \( S(m,k) \) in stage (4). (stages 4, 5 refers to the spectral analysis in Fig.(4.9))

Stage (6) Compute the root of the spectra computed in stage(5):

\[
S_\gamma(m,k) = |S(m,k)|^{\gamma}, \quad -1 \leq \gamma \leq +1
\]  

Stage (7) Take two-dimensional inverse discrete Fourier transform from the \( S_\gamma(m,k) \):

\[
\hat{c}(u,v) = \frac{1}{NM} \sum_{m=0}^{M-1} \sum_{k=0}^{N-1} S_\gamma(m,k) e^{j2\pi ku/N} e^{j2\pi mu/M}, \quad 0 \leq u \leq N-1, \quad 0 \leq v \leq M-1
\]  

Now, the TDRC matrix is defined as the absolute value of every element in matrix \( \hat{c}(u,v) \) in step (4) e.g:

\[
c(i,j) = |\hat{c}(u,v)|
\]  

where \( i, j \) are the row and column number in TDRC matrix. In the last stage and following the magnitude operation a sub matrix is selected from the lower dimensions of the original matrix. This sub matrix contains a set of coefficients most useful for speech recognition. As with the TDC, the axis \( v \) is called quefrency and has time dimension. The axis \( u \) is called time frequency and has frequency dimension. Fig.4.5 shows the time-frequency representation of two words using the TDRC features as an example. As shown in this figure significant values of features are located in the low quefrency and low frequency region. In Fig.4.6 the three first columns and the three first rows of TDRC matrix are shown. It illustrates that when \( u \) and \( v \) are increasing the TDRC coefficient are decreasing. Since coefficient \( c(0,0) \) relate to the energy it is set to zero in these figures for clarity, but it should be excluded from the feature set during the recognition process to avoid the influence of speech signal.
4.5. Two-dimensional root cepstrum analysis

Figure 4.5: Time-frequency representation of word “She” using the TDRC features
4.5.2 Properties of TDRC

The TDRC method provides some advantages over other methods. These advantages can be summarised as follows.

Feature compression

According to the property of symmetry in the two-dimensional Fourier transform operation most of the significant features are concentrated in the lower quarter part of the TDRC matrix. Thus the coefficients which are located at the lower index portion of the two-dimensional root matrix are more significant than the others and contain rich information about the speech signal. Furthermore the $\gamma$ power in $S(m,k)^\gamma$, equation (4.14), where $|\gamma| < 1$, can be considered as the a compression of the dynamic range of the data. As a result, only some TDRC coefficients are necessary to as a feature matrix for use in the classification stage.

Aspects of computation

As this method doesn’t need any time normalisation and the patterns are compared by computing the distance between the reference and test pattern, the computation should be easier compared to other methods. Moreover the stage in computing the feature matrix is easier compared to the PLP method which is widely used in speech recognition systems.

Robustness to Noise

The TDRC features are more robust to noise than MFCC features because:

1. The log function which is sensitive to noise is replaced with the root function.

2. The higher order of TDRC, in which the noise component is included, are omitted during the process of producing the TDRC sub-matrix.
4.5. Two-dimensional root cepstrum analysis

Figure 4.6: The three first columns and rows of the TDRC matrix

Preservation of both static and dynamic features

The TDRC characterises the variations in each cepstral coefficient in both time and frequency thus, it can be considered as an illustration of both static and dynamic features contained in the speech signal. The equation 4.14 can be separated into two stages:

\[ c_i(m, v) = \sum_{k=0}^{N-1} S_i(m, k) e^{j2\pi kv/N} \]

and

\[ \bar{c}(u, v) = \sum_{m=0}^{M-1} c_i(m, v) e^{j2\pi mu/M} \]

In the above equation the \( v \)-th coefficients illustrates the root cepstral coefficient of frame \( m \) and for a fixed \( v \), \( \bar{c}(u, v) \) is the spectrum of \( c_1(m, v) \) and indicates the variation of \( u \)-th root cepstral coefficient along the frames. Therefore, in the TDRC matrix with components \( \bar{c}(u, v) \), the characteristics of cepstra are preserved along the axis \( v \) whereas the variations of the cepstral coefficients are expressed along the axis \( u \). In other words, the characteristics of the signal cepstrum or static features are preserved in one dimension, while the changes of the cepstrum or dynamic features are indicated in the
other dimension. As a result the TDRC region can be divided to two main parts, static and dynamic. Each region also breaks up in to sub-regions, long time variation and short time variation as shown in Fig. (4.7).

![Figure 4.7: Regions of two-dimensional root cepstrum](image)

**Adjustable \( \gamma \) parameter**

As mentioned before, the adjustable \( \gamma \) parameter can be used to optimise the feature extraction process and enhance the performance. It has been shown in [103] that optimised modelling of speech data can be achieved by using the one dimensional root cepstrum and varying the \( \gamma \) value, implying that there is an optimum specific value of \( \gamma \) which depends on each frame of speech.

Fig. (4.8) shows the compression effects of \( \gamma \) along \( u \) and \( v \) axis for the TDRC. This figure illustrates that by varying the \( \gamma \) value the compression along both axes is changed which this allows for the optimisation of the feature extraction process thus enhances the performance. As can be seen from this figure the compression effects along \( u \) axis is more than \( v \) axis. When \( \gamma=0.9 \) the compression is low but it is increased when \( \gamma \) is decreased to 0.1. It is noticed that in this case, the compression is so large that some information is lost. Therefore one of the interesting objectives should be to select the
best value for $\gamma$ to provide the best performance. The selection of the optimum value for $\gamma$ varies from one application to another.

![Figure 4.8: The compression effects of $\gamma$ along 'v' and 'u' axis](image)

### 4.6 Complex TDRC

To include more information about the signal corresponding to the phase, the complex two dimensional root cepstrum can be defined as follows:

$$c_c(u, v) = 2D - IFT\{S(m, k)^\gamma\}$$  \hspace{1cm} (4.18)

where $2D - IFT$ indicates the two dimensional inverse Fourier transform, $S(m, k)$ is the Fourier transform of $s(m, n)$, being at the $n^{th}$ sample in frame $m$ of the speech signal. This equation is expanded as follows:

$$c_c(u, v) = 2D - IFT\{|S(m, k)|^\gamma[\cos(\varphi(m, k))] + j\sin(\varphi(m, k))\}$$  \hspace{1cm} (4.19)

where $\varphi(m, k)$ is the phase of $S(m, k)$. The equation 4.19 may be modelled as Fig.4.9. As shown in this figure it is necessary to unwrap the phase before doing any further operations.
Chapter 4. Two-dimensional cepstrum analysis

4.7 Conclusion

In this chapter time-frequency or two dimensional features used to represent a speech signal as a feature extraction process, were discussed. As most feature extraction methods involve a one dimensional feature either time or frequency they omit important features for recognition which exist in both time and frequency. The two dimensional cepstrum (either TDC or TDRC) can simultaneously preserve various different types of information contained in the speech signal. It can represent static and dynamic features, as well as global and fine frequency structure. Three types of two dimensional cepstrum were illustrated in this chapter. The TDC and CTM are basically similar but there are some different between them. The TDC applied a two-dimensional FT whereas the CTM applied two dimensional DCT. The mel scale frequency is used in CTM while TDC used only linear scale.

There is one weakness in the TDC method, the log function is sensitive to noise. To overcome this, we introduce a novel method, the two dimensional root cepstrum. In addition to the feature compression, which is more than TDC, the TDRC has some advantages over the TDC. It has the advantage of an adjustable $\gamma$ parameter which can be used to optimise the feature extraction process, reducing the dimensions of the feature matrix and giving simple computation and robustness to noise. More details about
4.7. Conclusion

TDRC method, supported by experimental results, are discussed in chapter 5 and 6. At the end of this chapter we introduced the complex TDRC to achieve some information corresponding to the phase. Initial experiments to show the ability of TDC and TDRC as powerful compression algorithm have been carried out. The performance of two dimensional feature extraction for clean and noisy speech recognition are presented in the next chapter.
Chapter 5

Evaluation of the performance of Two dimensional cepstrums

5.1 Introduction

In the previous chapter, the two dimensional cepstrum (TDC) method was described. Moreover the two dimensional root cepstrum (TDRC) was introduced as a novel method to overcome the disadvantages of the TDC. The properties of the TDRC were also discussed in detail and some initial experiments were carried out to support the theoretical discussion. In this chapter the performance of the TDC and the TDRC are evaluated experimentally as efficient and simple feature extraction methods for speech recognition. Since, in the classifier stage, we need to know the probability distributions of the features, which are produced by our proposed methods, it is necessary to estimate the density function of features before carrying out further investigations. Thus, a single multi-variable Gaussian function is assumed to represent the distribution of two dimensional features and then a mixture Gaussian model (GMM) is assumed later in this chapter as an improved estimate of the density function of features. In the next section of this chapter, the experimental arrangement, the schemes for computing the TDC or TDRC features and the data bases which are used in the experiments are introduced. Section 3 describes the classification method which is used to classify the speech. A primary estimation of the pdf of the proposed features is illustrated in section 4. Sec-
tion 5 offers experiments on the TDC with a discussion about the effects of varying the parameters of the TDC on recognition accuracy. This is followed by a comparison between the one dimensional cepstrum and the TDC. Section 6 offers an evaluation of the performance of the TDRC. The results of proposed methods on noisy speech recognition are discussed in section 7. A comparison between the TDC and TDRC is presented in section 8. Section 9 presents an estimation of the density function of two dimensional features using the GMM. Finally a summary of this chapter is given in section 10.

5.2 Experimental arrangement

As we mentioned before in chapter 4, section 4.3.4, there are three possible schemes for computing the TDC (or TDRC) matrix. The one which we adopt here is the third method where, the TDC matrix is calculated using the entire utterance without any frame deletion and only one TDC matrix is computed for each utterance. Since the entire utterance is used, all the characteristics are properly preserved. Furthermore, in this method, each utterance is represented by one compact feature matrix which is extracted from the total TDC matrix rather than a string of feature vectors. As an example for an utterance of 2673 samples in length (word 'she') and assuming a frame length of 256 samples, then a conventional MFCC would require 10 vectors each containing typically 26 elements (10 × 26). In the case of the TDC, there is one sub-matrix selected for the entire utterance containing typically just 20 × 3 elements.

Each utterance in the speech data is considered to have a fixed number of frames depending on its length. This number should be an integer power of two as appropriate for the FFT computation. If the length of the utterance is less than the corresponding allocated frame number, it is extended to the corresponding frame number by appending zeroes. Where the utterance has zeros appended, there may be some dilution of the information incorporated into the TDC. It is noted that this process is similar to that by [77]. Each frame size is set to 256 samples and the frames are spaced by 128 samples. To avoid the influence of speech energy, energy normalisation has been applied before any further processing. The total size of the feature matrix would be 16 × 256 or 32 × 256
5.3 Classification

etc. but only a sub-set is used to represent an utterance. The data base which we used in experiments is a set of selected words from the TIMIT database.

TIMIT data base

As a standard database, the TIMIT data base has been used. This is a fully labelled data base of Americans English which is created by Texas Instruments and the Massachusetts Institute of Technology. It consists of utterances of 360 speakers that represent the major dialects of American English. It is divided into eight dialect regions, with separate testing and training sections. Each speaker says eight different sentences as well as two common sentences. Since in this data base there are no common speakers it is completely speaker independent. For these experiments, in all, 418 words, uttered by 38 speakers, were selected from the TIMIT and used in training while 121 words, uttered by 11 different speakers, were used for testing recognition performance. The details of how this database was extracted are given in appendix A.

5.3 Classification

Statistical approaches have been used efficiently in pattern recognition. They can serve as good classification design in many applications such as speech recognition. In this thesis, the commonly used statistical approach, Bayes decision theory [24], is adopted and, as we will see, it can be simplified as a distance measure. The Bayes classification is the optimum classifier which minimises the probability of classification error [28] but its implementation is often difficult in practice because of its complexity, particularly when the dimensionality is high. The Bayes's rule expresses a posterior probability for a testing data $x$ as follows:

$$P(i|x) = \frac{p(x|i)P_i}{p(x)} , \quad i = 1, 2, \ldots, M_c$$  \hspace{1cm} (5.1)$$

where $P(i|x)$ is a posteriori probability of occurrence of class $i$ given that $x$ has already been observed. $P_i$ is probability of occurrence of class $i$, $p(x|i)$ the conditional class probability density function of $x$ given class $i$, $p(x)$ is the density function of $x$ and
Mc is the number of classes. The aim is to find the class index \( i \) which maximises $P(i|x)$. In Eq.(5.1) $p(x)$ is independent of $i$ and $P_i$ is assumed to be the same for all $i$, i.e all classes have the same probability distribution in database. Thus in order to maximise $P(i|x)$, we can just maximise $p(x|i)$. Therefore, for a given test vector $x$ to be recognised, the class index \( i_{\text{max}} \) is selected whose conditional probability $p(x|i)$ is the highest:

$$i_{\text{max}} = \arg \max [p(x|i)] \quad , i = 1, 2, ..., M_c$$

(5.2)

By assuming one Gaussian function for features of each class, the conditional probability is calculated as follows:

$$p(x|i) = p(x|\mu_i, C_i) = \frac{1}{\sqrt{(2\pi)^N |C_i|}} e^{-\frac{1}{2}((x-\mu_i)C_i^{-1}(x-\mu_i))}$$

(5.3)

where $N$ is the dimension of $x$, $\mu_i$ is the mean vector and $C_i$ covariance of $i$th class.

By using the training data of the $i$th class, we can estimate its mean vector $\mu_i$ and covariance matrix $C_i$.

### 5.3.1 Distance classification derived from the Bayes's rule

Applying a logarithm transformation and eliminating terms that are constant across words, i.e have a common $C$, eq.(5.3) reduces to the following:

$$\ln[p(x|\mu_i, C)] \propto (x - \mu_i)^T C^{-1} (x - \mu_i)$$

(5.4)

where $C$ is the covariance matrix for all classes. The right hand side of the proportionality in (5.4) is the Mahalanobis distance [51] which is widely used as a similarity measure for classification purposes. Therefore the classification based on Bayes rule and by maximising $p(x|\mu_i, C_i)$, reduces to the minimising of the Mahalanobis distance. Thus, the Mahalanobis measure can be used as a simplified form of the Bayes classifier to recognise a test vector. It can be defined for the distance between a test and reference pattern as follows:

$$d(x_t, x_r) = (x_t - x_r)^T C^{-1} (x_t - x_r)$$

(5.5)

where $C$ is the covariance matrix of total reference data and $x_r$ and $x_t$ are the reference and the test vector (the feature matrix is covet to a vector feature) respectively. When
5.4. Estimation of the Density Function of Features

$C^{-1}$ is equal to the identity matrix, the right part of Eq.(5.5) will reduce to the square Euclidean distance.

$$d(x_t, x_r) = (x_t - x_r)^T (x_t - x_r)$$ \hspace{1cm} (5.6)

A system based on this distance measure is used to classify the feature patterns. During the recognition process a test pattern is compared to all the reference vectors which are created based on the mean of classes by computing the distance of this test pattern to all the reference vectors. The decision rule is to find the reference vector which gives the minimum distance $d(x_t, x_r)$.

5.3.2 Other types of distance classifier

To design a good classification and find the best and simplest classification we also used two other types of distance classifier. The first one is the first order of Minkowski measure which can be defined as follows:

$$d(x_t, x_r) = \sum_{i=0}^{m} |x_r(i) - x_t(i)|$$ \hspace{1cm} (5.7)

where $x_t$ and $x_r$ are the reference and the test vectors respectively and $m$ is the dimension of the vectors.

The second one is defined as follows:

$$d(x_t, x_r) = |x_t||1 - x_t^T x_r/|x_t||x_r|$$ \hspace{1cm} (5.8)

which is suggested by Mansour and Juang [53] and we will call it the Mansour measure, $x_r$ and $x_t$ are the reference and the test vectors respectively and $|$ denotes the absolute values of vector.

These types are implemented and compared for performance in later sections of this chapter and next chapters.

5.4 Estimation of the Density Function of Features

It would be useful to know whether the proposed features are distributed normally. Probability density function estimation plays an important role in statistical pattern
recognition. There are three major approaches to density estimation (or probabilistic
modelling): parametric methods, semi-parametric and non-parametric methods. In
the parametric method, the probability density function is assumed to have a certain
probabilistic structure. In such cases, only the parameters of the pdf need to be esti­
mated. On the other hand, in non-parametric modelling, no model structure is assumed
and the pdf is directly estimated from the training data. Semi-parametric or mixture
modelling techniques are alternative approaches which offer a successful compromise
between parametric and non-parametric methods. In these methods the number of
free parameters are allowed to vary but a finite mixture of functions is assumed as the
functional form.

Here, we assume that the pdf of the proposed features is Gaussian as a general assump­
tion for unknown data (parametric method). It is necessary to have an idea of how
good this assumption is, since a multi-variable pdf is difficult to visualise. While the
features in the feature space are assumed independent, we can compile the histogram
of values for each feature derived from the entire database, to give an idea about the
pdf. Fig.5.1 and Fig.5.2 show the histograms of the four selected features taken from
the feature vectors for both TDC and TDRC methods . As shown in these figures,
the pdf of some of features appear almost Gaussian but not exactly a single Gaussian
distribution.

In addition to histograms, as a classical measure of a Gaussian fit, kurtosis is computed
for the features [26] to give a numerical measure of the pdf of features. Kurtosis
describes the peakedness of a distribution relative to the normal distribution. If the
distribution has a flattened shape (such as uniform distribution) its kurtosis is less than
3. On the other hand, if the distribution has a sharp peak near the mean with heavy
tails (such as Laplace distribution ) then its kurtosis is more than 3. The kurtosis of a
Gaussian random variable is 3.

Kurtosis for a random variable of $x$ is defined by the fourth moment divided by squared
variance as follows:

$$\beta = \frac{\mu_4}{\mu_2^2}$$ (5.9)
5.4. Estimation of the Density Function of Features

where \( \mu_i \) denotes the \( i \)th moment of \( x \):

\[
\mu_i = E[(x - \mu)^i]
\]  \hspace{1cm} (5.10)

and \( \mu \) is the mean value of \( x \) also known as the first raw moment.

Table (5.1) shows the kurtosis for feature vectors for all samples which have been produced by TDC and TDRC. As can be seen from this table, the mean of kurtosis for all features is close to 3 for both TDC and TDRC but the TDRC features are the closest to a Gaussian distribution.

5.4.1 Correlation matrix for two dimensional features

The conventional classifier needs a vector to use as a feature while our features are located in a matrix. Thus, it is necessary to convert the matrices to the vectors. But the question is how we can do it? To convert a matrix to a vector there are two easy ways. The first way is to split the matrix along the rows and put every column
Table 5.1: Kurtosis of Features both TDC and TDRC methods

<table>
<thead>
<tr>
<th>Feature number</th>
<th>Kurtosis TDC method</th>
<th>Kurtosis TDRC method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.47</td>
<td>2.71</td>
</tr>
<tr>
<td>2</td>
<td>2.61</td>
<td>2.62</td>
</tr>
<tr>
<td>3</td>
<td>2.11</td>
<td>2.32</td>
</tr>
<tr>
<td>4</td>
<td>2.46</td>
<td>2.03</td>
</tr>
<tr>
<td>5</td>
<td>2.29</td>
<td>2.19</td>
</tr>
<tr>
<td>6</td>
<td>2.46</td>
<td>2.47</td>
</tr>
<tr>
<td>7</td>
<td>2.43</td>
<td>2.73</td>
</tr>
<tr>
<td>8</td>
<td>2.38</td>
<td>2.69</td>
</tr>
<tr>
<td>9</td>
<td>2.05</td>
<td>3.17</td>
</tr>
<tr>
<td>10</td>
<td>2.40</td>
<td>2.62</td>
</tr>
<tr>
<td>11</td>
<td>2.42</td>
<td>2.64</td>
</tr>
<tr>
<td>12</td>
<td>2.17</td>
<td>2.60</td>
</tr>
<tr>
<td>13</td>
<td>2.5</td>
<td>2.87</td>
</tr>
<tr>
<td>14</td>
<td>2.40</td>
<td>2.86</td>
</tr>
<tr>
<td>15</td>
<td>2.38</td>
<td>2.76</td>
</tr>
<tr>
<td>16</td>
<td>2.23</td>
<td>2.85</td>
</tr>
<tr>
<td>17</td>
<td>2.22</td>
<td>2.94</td>
</tr>
<tr>
<td>18</td>
<td>2.32</td>
<td>2.82</td>
</tr>
<tr>
<td>19</td>
<td>2.44</td>
<td>2.85</td>
</tr>
<tr>
<td>20</td>
<td>2.22</td>
<td>3.03</td>
</tr>
<tr>
<td>Mean</td>
<td>2.35</td>
<td>2.69</td>
</tr>
<tr>
<td>Total mean</td>
<td>2.78</td>
<td>3.19</td>
</tr>
</tbody>
</table>
after the others in a string to make a vector. The second way is to split the matrix along the columns and put every row after the others. The first way can be done when the correlation between the coefficients of rows is not as strong as the correlation between the coefficients of columns. In contrast, the second way can be done when the correlation between the coefficients of columns is not as strong as the correlation between the coefficients of rows. Fig. 5.3 and Fig. 5.4 show the correlation matrices generated from the individual feature rows and columns for both TDC method and TDRC methods as an example only. The 2-D correlation matrix is computed by treating the rows and columns of data in the feature matrix as orthogonal and computing a sequence of 1-D autocorrelations. For instance, the procedure could start by taking each row of the feature matrix in term and computing an autocorrelation function for each row. In a $20 \times 20$ feature matrix, the result would be a $20 \times 40$ intermediate matrix consisting of 20 autocorrelation functions each of 40 autocorrelation values along the rows. Then a sequence of 1-D auto correlation calculations are made down the columns of this intermediate matrix generating the final $40 \times 40$ 2-D autocorrelation matrix.
Chapter 5. Evaluation of the performance of Two dimensional cepstrums

of the original data. Exactly the same result would be obtained if one started with
columns and followed this by calculating along the rows. As shown in both figures
the features are strongly decorrelated on one direction; when the column number is
increased (along the rows). It means that the coefficients along the columns are more
correlated than the coefficient along the rows. Therefore to convert the feature matrix
to a feature vector, it is better to split the matrix along the rows and this method was
adopted in our experiments.

Another observation which can be seen from these figure is that the two dimensional
methods produce the matrix in which most of features are decorrelated.

5.5 Experiments on the TDC

In order to demonstrate the efficiency of the TDC as a feature extraction method for
speech recognition several experiments are performed. This section describes various
experiments that were carried out in order to show the performance of the TDC feature
extraction method.
5.5. Experiments on the TDC

5.5.1 TDC performance

The recognition accuracies shown in table (5.2) give the results when conventional one dimensional cepstrum (ODC) and two dimensional cepstrum (TDC) are used as feature extraction methods in a recognition process. The database is a set of isolated words selected from TIMIT which were described in section (5.2). The classifier is the Mahalanobis distance for both ODC and TDC. The frame size for the ODC is the same as TDC, as we mentioned in section (5.2). Feature number for each word is assumed to be 20, 40, and so on as shown in the first column of table (5.2). For the TDC also the same number of features are assumed which are produced from sub-matrix of $20 \times 1$, $20 \times 2$, etc.. In the case of the TDC the recognition accuracy is increased when the feature number is increased but starts to decrease beyond 60 feature samples. However in the case of the ODC the accuracy increases almost linearly with number of features. Comparing the two columns of table (5.2), a noticeable improvement is shown by using the TDC. Moreover as shown in Fig.5.5 features which are extracted from the TDC method provide much better recognition accuracy than that of the conventional ODC for all numbers of features.
Chapter 5. Evaluation of the performance of Two dimensional cepstrums

Figure 5.5: The performance of ODC and TDC methods on recognition accuracy

Table 5.2: Recognition accuracy using ODC and TDC method (in % )

<table>
<thead>
<tr>
<th>Feature Number</th>
<th>ODC method</th>
<th>TDC method</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>55.4</td>
<td>73.6</td>
</tr>
<tr>
<td>40</td>
<td>58.2</td>
<td>83.6</td>
</tr>
<tr>
<td>60</td>
<td>63.6</td>
<td>85.4</td>
</tr>
<tr>
<td>80</td>
<td>67.3</td>
<td>79.1</td>
</tr>
<tr>
<td>100</td>
<td>63.6</td>
<td>75.9</td>
</tr>
<tr>
<td>120</td>
<td>65.4</td>
<td>74.5</td>
</tr>
<tr>
<td>140</td>
<td>67.3</td>
<td>75.4</td>
</tr>
<tr>
<td>160</td>
<td>66.4</td>
<td>76.4</td>
</tr>
<tr>
<td>180</td>
<td>70.0</td>
<td>73.6</td>
</tr>
</tbody>
</table>
5.5. Experiments on the TDC

Effects of removing DC-offset

For most of the speech data used here, the direct current (DC) or zero frequency is not zero. This is called DC-offset. Since this term is a common term in the speech wave, in our experiments we remove the DC-offset of each frame. As a result, the performance improved for the same size of TDC matrix. Fig.5.6 and tables (5.3), (5.4) show the effects of removing DC-offset on recognition accuracy.

5.5.2 TDC feature matrix dimensions

The dimension of TDC matrix has a great influence on recognition accuracy and computational complexity. Selecting less features or reducing the TDC feature matrix as much as possible makes the classifier easier and improves the recognition accuracy. Due to properties of symmetry in the DFT operation, all of the information corresponding to the signal is located in a quarter of the TDC matrix and normally a sub-matrix of the TDC is used for the feature matrix in the process of recognition. In the following, the effects of changing row, column and selecting optimum feature dimension for TDC sub-matrix are discussed.

Effect of changing rows

Table (5.3) shows the effect of changing the row number on recognition accuracy, before and after removing the DC-offset from the speech wave. In this table, the number of columns is fixed at 20. As can be seen from this table, if only the coefficients of the first row are used, the recognition rate is 35.6%. When the coefficients of the second rows are included in the feature matrix, the recognition rate increases to 40.0% (the second column of table i.e. in the case where the DC-offset is removed). By increasing the number of rows, the performance is further improved until a peak is reached at 8 x 20 elements of the TDC sub-matrix, with 77.3% accuracy. After that, when more rows are included, the performance worsens. Observing the 3-D diagram of TDC (chapter 4, Fig.4.1) and considering the properties of the two-dimensional Fourier transform, we find that there are few significant features at high row numbers. More details can be found in Fig.5.6.
Chapter 5. Evaluation of the performance of Two dimensional cepstrums

<table>
<thead>
<tr>
<th>TDC Sub-matrix</th>
<th>Recognition Rate For TDC method</th>
<th>Recognition Rate After removing DC-offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>1x20</td>
<td>35.6</td>
<td>35.6</td>
</tr>
<tr>
<td>2x20</td>
<td>37.1</td>
<td>40.0</td>
</tr>
<tr>
<td>3x20</td>
<td>52.4</td>
<td>56.7</td>
</tr>
<tr>
<td>4x20</td>
<td>66.5</td>
<td>68.2</td>
</tr>
<tr>
<td>5x20</td>
<td>66.5</td>
<td>68.2</td>
</tr>
<tr>
<td>6x20</td>
<td>68.7</td>
<td>70.1</td>
</tr>
<tr>
<td>7x20</td>
<td>69.5</td>
<td>72.8</td>
</tr>
<tr>
<td>8x20</td>
<td>73.2</td>
<td>77.3</td>
</tr>
<tr>
<td>9x20</td>
<td>69.6</td>
<td>72.3</td>
</tr>
<tr>
<td>10x20</td>
<td>67.9</td>
<td>70.0</td>
</tr>
</tbody>
</table>

Table 5.3: The effect of changing the number of rows on Recognition accuracy

Effects of changing columns

When the number of columns is increased, the recognition accuracy also is increased but only for the first few columns of the TDC sub-matrix. Also in this case we assume that the number of rows is fixed at 20. As shown in table 5.4 the recognition accuracy is 73.6% when the first column only is included in the TDC sub-matrix. There is an increase of about 10% to reach 83.6% when the number of columns increases from one to two (after removing the DC-offset). From the table, it can be seen that when the third column is included in the feature matrix the recognition accuracy improves significantly. However, after that, the recognition accuracy starts to decrease when more coefficients from other columns are added to the TDC sub-matrix. This means that the coefficients of these three columns (or the first few columns) are sufficient to form an effective feature matrix for the recognition of each word from the database.

We tested whether the results obtained from adding more columns are significantly better than when using only one column using the statistical test described in [31] with a 95% confidence (see also appendix B). The results of this test are given in table 5.5. According to the statistical analysis (table 5.5), it can be seen that when the
coefficients of the third column are incorporated, the recognition accuracy improves significantly while, there is no significant improvement in increasing from one column to two columns. Moreover, the statistical test also shows that there is no significant improvement when more than three columns are included in feature sub-matrix. It is noticed that the test which we used here assumes that the errors are uncorrelated. But this is not really true. A suitable test which gives more accurate results is McNemar's test [31].

<table>
<thead>
<tr>
<th>TDC Sub-matrix</th>
<th>Recognition accuracy For TDC method</th>
<th>Recognition accuracy After removing DC-offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>20x1</td>
<td>73.6</td>
<td>73.6</td>
</tr>
<tr>
<td>20x2</td>
<td>82.4</td>
<td>83.6</td>
</tr>
<tr>
<td>20x3</td>
<td>83.5</td>
<td>85.4</td>
</tr>
<tr>
<td>20x4</td>
<td>79.1</td>
<td>79.1</td>
</tr>
<tr>
<td>20x5</td>
<td>75.9</td>
<td>75.9</td>
</tr>
<tr>
<td>20x6</td>
<td>73.2</td>
<td>74.5</td>
</tr>
<tr>
<td>20x7</td>
<td>74.7</td>
<td>75.4</td>
</tr>
<tr>
<td>20x8</td>
<td>75.3</td>
<td>76.4</td>
</tr>
<tr>
<td>20x9</td>
<td>71.4</td>
<td>73.6</td>
</tr>
<tr>
<td>20x10</td>
<td>68.3</td>
<td>70.1</td>
</tr>
</tbody>
</table>

Table 5.4: The effects of changing number of columns on Recognition accuracy

**Feature selection**

The main goal of feature selection is to select a subset of \( d \) features from the given set of \( D \) measurements, \( d < D \), without significantly degrading the performance of the recognition accuracy. Feature selection can be considered as a mapping from the \( n \)-dimensional space to a lower dimensional feature space. The mapping should be carried out without severely reducing the class separability. Since the TDC method represents the features of speech signals in a matrix form, two simple ways of feature selection will be to select a sub-matrix, by varying the number of rows or columns as
Chapter 5. Evaluation of the performance of Two dimensional cepstrums

<table>
<thead>
<tr>
<th>Adding columns</th>
<th>Statistical significant?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two columns compared to one column</td>
<td>No</td>
</tr>
<tr>
<td>Three columns compared to one column</td>
<td>Yes</td>
</tr>
<tr>
<td>Four columns compared to one columns</td>
<td>No</td>
</tr>
<tr>
<td>Five columns compared to one columns</td>
<td>No</td>
</tr>
<tr>
<td>Six columns compared to one columns</td>
<td>No</td>
</tr>
<tr>
<td>Seven columns compared to one columns</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 5.5: The Statistical analysis for column adding

Figure 5.6: The Effect of changing column number and row number on recognition accuracy before and after removing the DC-offset.
5.5. Experiments on the TDC

we have seen previously. Fig. 5.7 shows the effect of feature selection by row and column change. This figure shows that increasing the column number is more effective than increasing row number. In other words, a feature matrix with $20 \times 4$ dimension has better performance than feature matrix with $4 \times 20$ dimensions while feature numbers for both are the same.

![Figure 5.7: Comparison of the effects of feature selection by row and column changing on recognition accuracy](image)

5.5.3 Optimum frame length

Experiments were conducted to analyse the effect of frame length on performance. Larger frame lengths produced lower recognition accuracy although this was not found to be a general rule. As shown in Fig. 5.8, by increasing the frame length the recognition accuracy at first increased and reach a peak at frame length = 16 ms (256 samples), then started to fall when it reached a dip at frame length = 25 ms. It started to increase and reach a second peak at frame length = 32 ms (512 samples). After that it started to decline again until 54 ms. It reach to a third peak at 64 ms. As it is clear the highest peak is only at 16 ms. The main reason is related to the effect of zero padding. When the frame length is increased the number of zeros which are appended needs to be increased. Because of this in the process of computing the TDC, some features will be set to zero which causes the recognition accuracy to get worse. In contrast, when
Chapter 5. Evaluation of the performance of Two dimensional cepstrums

the frame size is so low (below 16 ms) the recognition accuracy is also low. Therefore an optimum frame length is needed to be selected.

![Figure 5.8: The effect of increasing frame length (TDC method)](image)

5.5.4 Classifier performance comparison

As we mentioned before, the principal aim of this thesis is feature extraction but in order to find a simple and well behaved classifier some experiments have been carried out to investigate the effects of distance classifier design on recognition accuracy. Fig.5.9 shows the results of classifier on recognition accuracy using the TDC features. The performance of four classifiers, namely: Mahalanobis: Mah, Mansour: Man, Minkowski: Mink and Euclidean: Euc. were considered. From Fig.5.9 it can be seen that the Mahalanobis distance classifiers provides the best results compared to the others. The Euclidean distance classifier gave worst performance.

5.5.5 Comparison of ODC and TDC

The performance of the TDC has been compared with the ODC using Mahalanobis distance on section (5.5.1). Now here, more details are presented using all four types of classifiers. Tables (5.6) and (5.7) show the results of ODC and TDC for different types of classifier. By comparing these tables it is clearly observed that the recognition accuracy
5.5. Experiments on the TDC

Figure 5.9: A comparison of recognition accuracy for four classifiers (TDC method)

of TDC method for all classifiers is much better than that of the ODC. Moreover the maximum recognition accuracy of the ODC is 67.3 % which corresponds to 80 features and the Mahalanobis classifier. However the recognition accuracy is 85.4 % for the TDC method which corresponds to 60 features and the same distance classifier. This means that using the TDC method not only increases the recognition accuracy but also reduces the optimum feature number.

Table 5.6: Recognition accuracy( in % ) for various feature numbers and classifier using ODC

<table>
<thead>
<tr>
<th>Feature Number</th>
<th>ODC</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>26.4</td>
</tr>
<tr>
<td>40</td>
<td>29.1</td>
</tr>
<tr>
<td>60</td>
<td>31.8</td>
</tr>
<tr>
<td>80</td>
<td>38.1</td>
</tr>
<tr>
<td>100</td>
<td>40.7</td>
</tr>
</tbody>
</table>
Table 5.7: Recognition accuracy (in %) for various feature numbers and classifier using TDC

<table>
<thead>
<tr>
<th>Sub Matrix</th>
<th>Feature Number</th>
<th>TDC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>TDC</td>
</tr>
<tr>
<td>20×1</td>
<td>20</td>
<td>46.4 58.1 48.1 73.6</td>
</tr>
<tr>
<td>20×2</td>
<td>40</td>
<td>51.8 60.1 56.3 83.6</td>
</tr>
<tr>
<td>20×3</td>
<td>60</td>
<td>50.0 59.0 60.0 85.4</td>
</tr>
<tr>
<td>20×4</td>
<td>80</td>
<td>51.8 63.6 61.8 79.1</td>
</tr>
<tr>
<td>20×5</td>
<td>100</td>
<td>50.0 62.7 58.6 75.4</td>
</tr>
</tbody>
</table>

5.6 Experiments on the TDRC

This section describes various experiments that were carried out in order to show the performance of TDRC feature extraction method on recognition accuracy for speech. Several experiments were performed to analyse the effects of the TDRC method and the variation of corresponding parameters, on the performance of recognition, especially the effect of changing $\gamma$. A comparison between TDC and TDRC is illustrated on section (5.8).

As we mentioned in chapter 4 the TDRC has some advantages over the TDC. Therefore it is expected that the TDRC will offer a better performance compared to the TDC in this application.

5.6.1 TDRC feature matrix dimensions

As with the TDC, the TDRC represents any utterances in a matrix form. Also, the dimensions of TDRC have a great influence on the recognition accuracy and computational complexity. Choosing proper dimensions for the feature matrix, which will maximise the recognition accuracy, is crucial. It is important to investigate how many rows and columns should be selected as a feature matrix, to have the best representation of any utterance. Moreover, it is noticed that the optimum dimension of TDRC
5.6. Experiments on the TDRC

(or TDC) are highly dependent on the word length and the word identity. However we can estimate an optimum dimension for a set of words. In the following, again the effects of changing row, column and selecting optimum feature dimension for TDRC are discussed. It should be noticed here that as the TDRC coefficients depend on \( \gamma \) therefore for each value of \( \gamma \) we will have different optimum dimensions of the TDRC sub-matrix.

Effects of changing rows and columns

Fig.5.10 shows the effects of row and column change on recognition accuracy for different values of \( \gamma \). In the left hand figure, the effects of row number for a fixed number of columns is shown while in the right hand figure, the effect of changing columns for a fixed number of rows is shown. It can be seen from the figure that to select a sub-matrix the number of rows which is needed is more than 10 rows while the column number should be less than 5. Also it is shown that the number of rows or columns which are needed to achieve the maximum accuracy is dependent on the \( \gamma \) value. Another observation from the figure is that the curve which corresponds to \( \gamma = 0.3 \) has the better performance compare to other values of \( \gamma \). It is noticed that due to compression effects of \( \gamma \), the optimum sub-matrix is changed by the \( \gamma \) value. As, it was shown in chapter 4, the compression effect of \( \gamma \) in \( u \) axis (or column) is much bigger than on the \( v \) axis. Thus, to form a TDRC sub-matrix we need to select a higher row number than column number. In summary, as with the TDC method, in order to form an effective matrix to represent each word of the database, it is necessary to include only the coefficients of the first few columns with the coefficients of more than 10 rows.

5.6.2 Optimum frame length

According to our experiments the effects of frame length in TDRC feature is almost the same as TDC as shown in Fig.5.11. As shown in this figure, by increasing the frame length the recognition accuracy at first increased and reached a peak at 16 ms, the same frame length of TDC, then started to fall when it reached a dip at frame length=32 ms. It started to increase and remained constant until a length of 50 ms. After that it started to decline slightly.
Figure 5.10: Effects of row and column changing on recognition accuracy for TDRC method

Figure 5.11: The effect of increasing frame length (TDRC method)
5.6.3 Effect of $\gamma$ variation

As we mentioned in chapter 4, one of the advantages of the TDRC is the parameter $\gamma$ that can be used to maximise the recognition accuracy, which is not available in the TDC. As the range of $\gamma$ is assumed to be between $-1$ and $+1$ (see chapter 4 definition of TDRC) we consider four regions for $\gamma$. We should mention that when $\gamma$ is close to 0, the root cepstrum converges to log cepstrum [48], thus we are not interested in this as it will produce the same result as the log cepstrum. These regions are as follows:

1. $\gamma < -1$
2. $\gamma > 1$
3. $-1 < \gamma < 0$
4. $0 < \gamma < 1$

Fig.5.12 shows the effect of $\gamma$ variation on recognition accuracy when the feature matrix is $20 \times 3$ for both regions (1) and (2). The classifier which is used here is the Mahalanobis which offers the best results. As can be seen from this figure, very poor performance is achieved in these two regions. In summary extending $\gamma < -1$ or $\gamma > 1$ leads to unsatisfactory accuracy and very poor results.

Fig.5.13 shows the effects of $\gamma$ variation for both regions (3) and (4). For negative values the recognition accuracy is increased when the $\gamma$ value is increased, as $\gamma$ tend to 0. On the other hand for positive values the recognition accuracy starts to increase when $\gamma=0.1$ and reaches a global maximum after which it starts to fall. When $0.2 < \gamma < 0.4$ the recognition accuracy of the TDRC method is higher than the recognition accuracy for the TDC. As shown in chapter 4, when $\gamma$ is small the compression effects are high which means that some information is lost. Thus for small value of $\gamma$ the recognition accuracy is not high. Also, when $\gamma$ is high the compression is low and once again the recognition accuracy is low because high row and column numbers are needed to represent the utterance. As shown in the figure the optimum value which maximises the accuracy is around $\gamma=0.3$. This agrees closely with observations made in [103]. In addition, in figure (5.14) the recognition accuracy, as a function of $\gamma$ and column number
(which is more important than row number) when number of rows are fixed, is shown. Fig.5.14.a depicts the negative values of $\gamma$ while Fig.5.14.b depicts the positive values of $\gamma$. It can be seen that by varying $\gamma$ and adjusting the column number, maximum performance can be attained while row number is fixed.

![Graphs showing performance variation with $\gamma$.](image)

Figure 5.12: The effects of $\gamma$ variation when $\gamma$ is less than $-1$ (left) and more than 1 (right)

Figure 5.13: The effects of $\gamma$ variation when $-1 < \gamma < 0$, left, and $0 < \gamma < 1$, right

5.6.4 A comment on performance optimisation

The optimisation of the feature matrix was based on a relatively small database. Ideally this process should have been carried out on a larger database to ensure optimum
5.7. Noisy speech recognition

Extracting robust features for noisy speech is crucial. Speech recognition in adverse environments is one of the major issues in automatic speech recognition nowadays. While most current speech recognition systems are highly efficient for ideal environments, their performance degrades extremely when they are applied to real environments because noise distorts speech. The problem of speech recognition in noisy environments has attracted the interest of many researchers. The noise reduction method for speech

performance. This is left for future investigation.

5.6.5 Classifier performance comparison

Fig. 5.15 shows results of different types of distance classifier when the TDRC features are used. It can be seen from this figure that the Mahalanobis provides the best performance. After that the Mansour distance offers better performance while the Euclidean distance classifier gave the worst performance. This is similar to the results found for the TDC, section (5.5.4).

5.7 Noisy speech recognition

Figure 5.14: Effect of changing $\gamma$ on recognition accuracy.

(a) Negative $\gamma$, $-1 < \gamma < 0$

(b) Positive $\gamma$, $0 < \gamma < 1$
recognition system can be divided into three important categories. The first approach is the use of features which are more robust to noise \[23\] \[2\], the second approach is compensation and filtering noisy speech \[10\] and the third is adaptation of the speech models to include the effects of noise \[98\] \[94\]. A detailed survey of noisy speech recognition techniques can be found in \[32\]. Our proposed method for noisy speech recognition is concerned with the first approach.

The noisy utterance is simulated by adding artificially generated white Gaussian noise to the clean speech signal with various SNR levels by the following equations:

\[
\hat{s}[m] = s[m] + kn[m] \tag{5.11}
\]

\[
k = \frac{\sum_{m=0}^{M-1} s^2[m]}{10^{\frac{SNR}{10}} \sum_{m=0}^{M-1} n^2[m]} \tag{5.12}
\]

\[
SNR = 10 \log \frac{\sum_{m=0}^{M-1} s^2[m]}{\sum_{m=0}^{M-1} n^2[m]} \tag{5.13}
\]

where $\hat{s}$ and $s$ and $n$ represent noisy speech signal, clean speech signal and noise signal respectively, $M$ is the length of $s$, and SNR denotes the signal to noise ratio.

Some experiments were performed to investigate the effect of the suggested method on recognition accuracy in noisy conditions. Figure 5.16 shows the results of recognition accuracy of using TDC features and TDRC features in noisy conditions when the feature
sub-matrix is $20 \times 3$. For TDRC, $\gamma = 0.31$ gives the best performance. To investigate whether any apparent difference in performance of TDC and TDRC is statistically significant we also used the statistical test described in [31]. The results of this test are given in table 5.8.

<table>
<thead>
<tr>
<th>Comparison of TDC and TDRC in noisy conditions</th>
<th>Statistical significant?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result of comparison at SNR=34 dB</td>
<td>Yes</td>
</tr>
<tr>
<td>Result of comparison at SNR=28 dB</td>
<td>Yes</td>
</tr>
<tr>
<td>Result of comparison at SNR=22 dB</td>
<td>No</td>
</tr>
<tr>
<td>Result of comparison at SNR=16 dB</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 5.8: The Statistical analysis of a comparison of TDC and TDRC in noisy conditions.

As can be seen, with the same condition, the TDRC provides statistically better performance than original TDC for some levels of SNR.
Chapter 5. Evaluation of the performance of Two dimensional cepstrums

5.8 Comparison of TDC and TDRC

Fig. 5.17 shows the results of recognition accuracy when the number of rows and columns are changed for both TDC and TDRC methods. As can be seen from the figure the recognition accuracy is much larger for TDRC in (5.17.a) and (5.17.b) when the number of rows and column are changing. Also table (5.10) shows the result of using TDRC features for different types of distance measures and size of sub-matrix. In table(5.9), the result for TDC is repeated here again for convenience to make the comparison clearer. By comparing table(5.10) and table(5.9) it can be seen that the TDRC method offers a better accuracy than the TDC for all types of distance measure and sub-matrixes size. Moreover, as shown in table(5.9) a sub-matrix of 20 × 3 presents 84.4% accuracy while the TDRC method gives more than this amount with only a 20 × 2 sub-matrix. This means that by using a vector size of 40 feature numbers we can achieved an accuracy of 86.5% from TDRC method whereas TDC method offer an accuracy of 85.4% with 60 feature numbers. Thus the TDRC present significant reduction in the size of feature matrix. Once again it should be noticed that this is achieved by varying the γ and finding the optimum value. These results demonstrate the outstanding capability of TDRC [58].
### 5.8. Comparison of TDC and TDRC

Table 5.9: Recognition accuracy (in %) for various feature numbers and classifier using TDC (repeated of table 5.7)

<table>
<thead>
<tr>
<th>Sub Matrix</th>
<th>Feature Number</th>
<th>TDC</th>
</tr>
</thead>
<tbody>
<tr>
<td>20×1</td>
<td>20</td>
<td>46.4</td>
</tr>
<tr>
<td>20×2</td>
<td>40</td>
<td>51.8</td>
</tr>
<tr>
<td>20×3</td>
<td>60</td>
<td>50.0</td>
</tr>
<tr>
<td>20×4</td>
<td>80</td>
<td>51.8</td>
</tr>
<tr>
<td>20×5</td>
<td>100</td>
<td>50.0</td>
</tr>
</tbody>
</table>

Table 5.10: Recognition accuracy (in %) for various feature numbers and classifier using TDRC

<table>
<thead>
<tr>
<th>Sub Matrix</th>
<th>Feature Number</th>
<th>TDRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>20×1</td>
<td>20</td>
<td>51.8</td>
</tr>
<tr>
<td>20×2</td>
<td>40</td>
<td>60.0</td>
</tr>
<tr>
<td>20×3</td>
<td>60</td>
<td>62.7</td>
</tr>
<tr>
<td>20×4</td>
<td>80</td>
<td>63.6</td>
</tr>
<tr>
<td>20×5</td>
<td>100</td>
<td>60.0</td>
</tr>
</tbody>
</table>
5.9 Gaussian mixture model for features

In section (5.4) we considered the pdf of our proposed features as a single multi variable Gaussian function. As we have shown, the pdf of features is close to a Gaussian function but it is not exactly a single Gaussian function. Now in this section a semi-parametric or Gaussian mixture model is assumed to represent the distribution of two dimensional features and the results are compared with the single Gaussian. The history of mixture modelling started over 100 years ago. Pearson [84] used a mixture of two Gaussian functions for modelling a data set. The most attractive density function which is used widely is the Gaussian function. However other functions have been shown to be effective in special applications [69]. Nowadays, finite mixture models are widely used in different fields and applications specific to speech recognition, speaker identification and speaker recognition. In speech recognition, GMM are often used to model the state distribution of Hidden Markov Models (HMM). Reynold [96] applied the mixture models to speaker identification tasks. A speech recognition system based on density mixture Gaussian pdfs often demonstrates a better speech recognition performance than using a single Gaussian pdf. However, one of the problems associated with using GMM in speech recognition is its computational cost. Evaluating a mixture Gaussian pdf in speech recognition consumes significant amount of CPU time [96].

In the following, the GMM is described briefly. To estimate the GMM parameters, the expected maximum (EM) algorithm also will be illustrated and at the end of this section the results of using GMM as estimated density functions of features are presented.

A Gaussian mixture model(GMM) is a weighted sum of several multivariate Gaussian densities, as depicted in Fig. 5.18 and is given by:

$$p(x) = \sum_{j=1}^{M_g} p(x|j)P_j$$

(5.14)

where $p(x|j) = G(x|\mu_j, C_j)$ is a Gaussian family of pdfs:

$$G(x|\mu_j, C_j) = \frac{1}{\sqrt{(2\pi)^N|C_j|}}e^{-\frac{1}{2}[(x-\mu_j)^T C_j^{-1}(x-\mu_j)]}$$

(5.15)

$\mu_j$ is a mean vector, $C_j$ is a covariance matrix, $N$ dimension of $x$ and $M_g$ is the total
5.9. Gaussian mixture model for features

Figure 5.18: Depiction an M component Gaussian mixture model

number of components. The coefficient $P_j$ is a mixing parameter and represents the weight associated with component function, $p(x|j)$. They are chosen such that:

$$\sum_{j=1}^{M_g} P_j = 1 \quad \text{and} \quad 0 < P_j < 1 \quad (5.16)$$

Thus, a Gaussian mixture model presents each class of data as a linear combination of several Gaussian pdfs in the features space. The complete Gaussian model is parameterised by the mean vectors, covariance matrices and mixture weights from all component densities. These parameters are collectively represented by the notation:

$$\lambda = \{\mu_j, C_j, P_j, \} \quad j = 1, ..., M_g \quad (5.17)$$

5.9.1 GMM parameters estimation

Given the number of components, $M_g$, the aim is to estimate the GMM parameters, i.e $\lambda = \{\mu_j, C_j, P_j, \}$. During the training process, the maximum likelihood (ML) estimation is applied to determine the model parameters which maximise the likelihood of the GMM given the training data. Suppose that $X_{N_1} = \{x_1, x_2, ..., x_{N_1}\}$ is a set of training sample data, the GMM likelihood of $\lambda$ with respect to the set of training data
vector $x_i$ is defined as:

$$L(\lambda) = \prod_{i=1}^{N_t} p(x_i|\lambda)$$  \hspace{1cm} (5.18)

where $N_t$ is number of training sets. The optimal parameters can be obtained by maximising the equation (5.18). Since the logarithm function increases monotonically, in practice the negative log-likelihood of $L(\lambda)$ is minimised instead of maximising $L(\lambda)$.

$$E(\lambda) = - \ln \{ L(\lambda) \} = - \sum_{i=1}^{N_t} \ln \{ p(x_i|\lambda) \} = - \sum_{i=1}^{N_t} \ln \left\{ \sum_{j=1}^{M_g} P_j G(x_i|\mu_j, C_j) \right\}$$  \hspace{1cm} (5.19)

The $E(\lambda)$ is minimised with respect to $\lambda$ by setting:

$$\frac{\partial E(\lambda)}{\partial \lambda_j} = 0 \hspace{1cm} \forall j \in [1, ..., M_g]$$  \hspace{1cm} (5.20)

The analytical solution of (5.20) leads to a set of nonlinear equations. Therefore there needs to be a numerical solution to find the $\lambda_{opt}$. The Expectation Maximisation (EM) algorithm is a standard solution to this problem [1]. In the following section, this method is described briefly.

The EM algorithm

The EM algorithm is an iterative method which begins with an initial model $\lambda$ and aims at estimating a new model $\tilde{\lambda}$ such that $E(\lambda) \leq E(\tilde{\lambda})$ or $L(\lambda) \geq L(\tilde{\lambda})$. Since each iteration of the algorithm consists of an expectation step (E-step) followed by a maximisation step (M-step) it is called the EM algorithm. The general EM algorithm can be described in the following way:

1. Initialisation: Choose an initial estimate $\lambda = \{\mu_j^{(0)}, C_j^{(0)}, P_j^{(0)}\}$

2. E-step: In this step the posterior probability that component $j$ is responsible for the generation of pattern $x_i$ is estimated based on the current parameters, i.e.

$$p^{(l)}(j|x_i) = \frac{P_j G(x_i|\mu_j, C_j)}{\sum_{j=1}^{M_g} P_j G(x_i|\mu_j, C_j)}$$  \hspace{1cm} (5.21)

where $(l)$ denotes the value of the parameters at the $l^{th}$ iteration.
3. M-step: In this step the estimate of the new parameter is obtained. The mixture weight is obtained using the following formula:

$$
P_j^{(t+1)} = \frac{1}{N_t} \sum_{i=1}^{N_t} p_j^{(t)}(j|x_i)
$$

(5.22)

The mean vectors and covariance matrix are obtained by the following formulas respectively.

$$
\mu_j^{(t+1)} = \frac{1}{N_t} \sum_{i=1}^{N_t} p_j^{(t)}(j|x_i)x_i
$$

(5.23)

$$
C_j^{(t+1)} = \frac{1}{N_t} \sum_{i=1}^{N_t} p_j^{(t)}(j|x_i)(x_i - \mu_j^{(t+1)})(x_i - \mu_j^{(t+1)})^T \sum_{i=1}^{N_t} p_j^{(t)}(j|x_i)
$$

(5.24)

The update of equation (5.24) can be expressed in terms of current conditional expectation of $T_j^1$, $T_j^2$ and $T_j^3$, given by the following equation, which is computational efficient [62].

$$
T_j^1 = \sum_{i=1}^{N_t} p_j^{(t)}(j|x_i)
$$

(5.25)

$$
T_j^2 = \sum_{i=1}^{N_t} p_j^{(t)}(j|x_i)x_i
$$

(5.26)

$$
T_j^3 = \sum_{i=1}^{N_t} p_j^{(t)}(j|x_i)x_i^T x_i
$$

(5.27)

Now the equation 5.24 is written as

$$
C_j^{(t+1)} = \frac{T_j^3 - T_j^2 T_j^2^T}{T_j^1}
$$

(5.28)

These processes are iterated until either the log-likelihood value is small enough or a maximum number of iterations is performed.

### 5.9.2 Initialisation of GMM

The initialisation of GMM in the EM algorithm is important. The EM algorithm is guaranteed to find a local minimum log-likelihood model regardless of the initialisation, but different initialisations can lead to different local minima. The simplest way is to select the weights $P_j$ by $1/M$, where $M$ is the number of Gaussian components. The covariance matrices are initialised so that the diagonal elements are set at one and the rest are zero. The initialisation of mean vectors can either be placed by randomly chosen input vector or by using an alternative method, the k-means clustering algorithm.
K-Means clustering algorithm

The k-means clustering algorithm is an algorithm which a set of $L$ training vectors can be clustered into a set of $k$ clusters iteratively such that a distance measure keeps being minimised. Every class has its centre or mean vector and in a k-mean iteration every example vector is assigned to the class with the closest mean vector. After that every mean vector is replaced by the average of all vectors that have been assigned to it. The steps of the algorithm can be summarised as following:

1. Initialisation: Select arbitrarily $k$ clusters out of $L$ vectors and initialise the clusters centres or mean.

2. Nearest-Neighbour search: For each training vector, assign cluster labels by finding the closest cluster centres.

3. Update: Compute cluster centres for all cluster and reassign cluster labels.

4. Iteration: Repeat step 2 and step 3 until the average distance measure falls below a preset threshold [28].

5.9.3 Experiments with GMM

Classification process

It is noticed that here, when we assume a GMM model for features, we cannot use the distance classification that we used before (i.e Mahalanobis). Thus we apply directly the Bayes’ rule and then use the maximum likelihood approach (see e.q 5.1 and then e.q 5.2). This has been done by calculating the probability of observing a pattern given an estimated distribution of patterns for each class. During the training phase, a Gaussian mixture model is estimated to represent each class (or word). In testing, for a given test word, the recognition process is to find the class which gives the highest probability.
5.9. Gaussian mixture model for features

Results with GMM

Figure (5.19) shows the Gaussian Mixture approximation for the first feature of the TDC feature vector as an example and table (5.11) shows the mean square error between the approximation of the pdf by GMM and from the histogram for different component numbers. As shown in figure (5.19) the fit improves by increasing the components from two to five. A good fit appears for five components, but is poorer for six or more. Also, it can be seen from table (5.11) that minimum mean square error (mse) is 0.0015 which is corresponding to 5 Gaussian components and after that it increases when the component number is increased. Thus from this simple example, we can see that a set of Gaussian functions can be used as a better pdf estimation for all features. As in feature space we always have about 60 features for each word, it is necessary to find a set of 60 dimension mixture Gaussian components. It means that each feature vector word can be modelled by parameters of a set of mean vectors, covariance matrix and weight of each mixture components i.e $\lambda = \{\mu_j, C_j, P_j,\} j = 1, ..., M_g$. The covariance matrix is assumed to be a $60 \times 60$ diagonal matrix for each mixture component.

![GMM pdf estimation with 2 components](image1)
![GMM pdf estimation with 4 components](image2)
![GMM pdf estimation with 5 components](image3)
![GMM pdf estimation with 7 components](image4)

Figure 5.19: Gaussian Mixture approximation for the first feature of feature vector as an example.
Table 5.11: Mean square results

<table>
<thead>
<tr>
<th>Gaussian components</th>
<th>Mean square error</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.0030</td>
</tr>
<tr>
<td>3</td>
<td>0.0018</td>
</tr>
<tr>
<td>4</td>
<td>0.0018</td>
</tr>
<tr>
<td>5</td>
<td>0.0015</td>
</tr>
<tr>
<td>6</td>
<td>0.0017</td>
</tr>
<tr>
<td>7</td>
<td>0.0017</td>
</tr>
<tr>
<td>8</td>
<td>0.0016</td>
</tr>
</tbody>
</table>

Table (5.12) shows the results of assuming a Gaussian mixture as an estimation of our proposed features (TDC and TDRC) on recognition accuracy. As shown in table (5.12) the recognition accuracy is increased when the Gaussian components are increased and reach a maximum at 3 components. Thus the best performance is obtained by estimating 3 components for total of the feature vector.

We can see from this table that the GMM improved the recognition accuracy when compared to the one Gaussian mixture, but it has some disadvantages. First, it should be noted that in this case the classification stage cannot be simplified to a simple distance measure (as mentioned before). Second, increasing the number of components increases the size of coefficients which are used in the classification stage. Therefore these make the classification stage more complex. Third, the fundamental problem with the GMM is that the number of mixtures required to adequately describe the data is not known in advance and needs an investigation to find the optimum number. Yet another problem associated with using GMM its computational cost. Evaluating a mixture Gaussian pdf consumes significant amount of CPU time.

5.10 Summary and conclusion

In this chapter, the performance of TDC and TDRC have been evaluated experimentally for speech recognition. Experimental results illustrate that both the TDC and the
5.10. Summary and conclusion

<table>
<thead>
<tr>
<th>Gaussian components per word</th>
<th>Recognition accuracy (%)</th>
<th>Recognition accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TDC method</td>
<td>TDRC method</td>
</tr>
<tr>
<td>2</td>
<td>86.4</td>
<td>89.5</td>
</tr>
<tr>
<td>3</td>
<td>87.1</td>
<td>90.2</td>
</tr>
<tr>
<td>4</td>
<td>86.4</td>
<td>89.3</td>
</tr>
<tr>
<td>5</td>
<td>86.4</td>
<td>87.6</td>
</tr>
<tr>
<td>6</td>
<td>84.7</td>
<td>86.7</td>
</tr>
<tr>
<td>7</td>
<td>83.2</td>
<td>84.5</td>
</tr>
</tbody>
</table>

TDRC outperform the conventional cepstrum. It means that the speech characteristics are represented more accurately by two dimensional cepstrums (TDC and TDRC). We demonstrated that the effects of column change on recognition accuracy in both TDC matrix and TDRC is more effective than changing row number. In other words, to form an effective matrix to represent each word of the database, it is necessary to include only the coefficients of the first few columns with the coefficients of significantly more rows. We show that a feature matrix with $20 \times 3$ dimension has better performance than feature matrix with $3 \times 20$ dimensions while feature numbers for both are the same. It is shown that the TDRC outperforms the TDC while presenting significant reduction in the size of the feature matrix due to efficient compression ability of $\gamma$ over the two axes especially on the $u$ axis which is corresponding to the column. The superior performance of TDRC over TDC on noisy speech also discussed in this chapter. There are several reasons for the superior performance of TDRC over the TDC. Firstly this is related to the substitution of $\log$ function with $\sqrt{}$ function as mentioned in chapter 4, section(4.5). In the $\log$ function as argument, $x$, tends to zero, $\log(x)$ tends to minus infinity, thus $\log$ function is very sensitive to small values of $x$. In the TDC this means that there is more sensitivity to those parts with lower spectral power. Thus this makes its performance worse than TDRC. Secondly the optimisation $\gamma$, as already discussed, gives an improvement in the TDRC. Thirdly, the form of the pdf for the TRDC features is superior. As a result of our experiments the value of kurtosis for
Chapter 5. Evaluation of the performance of Two dimensional cepstrums

TDRC features is closer to 3 than for the TDC. This means that the TDRC pdf is nearer to Gaussian than the TDC. Hence in classification it will produce better results than TDC on recognition accuracy.

According to our experiments the optimum value for $\gamma$ is varied depending to the feature sub matrix, the type of distance classifier and the data. However this optimum value is around 0.3 for all of them. The superior performance of TDRC over TDC on noisy speech will be discussed in the next chapter.

Also, of the four types of distance classifier, the Mahalanobis classifier provides the best performance for both TDC and TDRC. The Mansour distance classifier offers less performance though it outperforms the remaining two for the TDRC. There is no significant difference between the Mansour measure and Mikowski distance for TDC features. The Euclidean distance classifier gave worst performance for both TDC and TDRC.

In the penultimate section of this chapter, the GMM was suggested to estimate the pdf of two dimensional feature. Although it improved the results, there was some difficulty in implementation. First, it should be noted that in this case the classification stage cannot be simplified to a simple distance measure. Thus the Basy's classification has been used. Moreover increasing the number of components increases the size of computation which is used in the classification stage. Therefore these make the classification stage more complex. Second, the fundamental problem with the GMM is that the number of mixtures required to adequately describe the data is not known in advance and needs an investigation to find the optimum number.
Chapter 6

Developing the proposed methods

6.1 Introduction

In the previous chapter we demonstrated empirically that the two dimensional cepstrum, especially the TRDC, can be used successfully as an efficient feature extraction method for the application to speech recognition. In this chapter the proposed methods are modified using different techniques. In the first step, linear discriminant analysis (LDA) is applied to the features to achieve extra dimensionality reduction while preserving as much of the class discriminatory information and improving the recognition accuracy [56]. At the next stage, the mel frequency scale is suggested to apply to the TDRC. Thus we introduce the Mel TDRC as a modified method to improve the accuracy. Since in most feature extraction methods, including the two dimensional cepstrum, the magnitude of the Fourier transform (FT) is used, information corresponding to phase is lost. In the next step, the Hartley transform (HT) is suggested as a substitute for the FT in two-dimensional cepstrum analysis to preserve both magnitude and phase details of the speech signal simultaneously.

The structure of this chapter is as follows. In section 7.2 we will review the main point of the LDA. In section 7.3 experimental results applying the LDA to both TDC and
TDRC are presented. Section 7.4 the Mel TDRC is introduced followed by experimental results with Mel scale frequency. Section 7.5 reviews the Hartley transform and Hartley cepstrum. The two dimensional Hartley root cepstrum is introduced in section 7.6 with the results of using this novel method. A comparison study of different types of feature extraction approaches for speech recognition is given in section 7.8. Finally a conclusion is given in section 7.9.

6.2 Main Point of LDA

The LDA introduced by Fisher [24] is a powerful tool for pattern recognition in general which can be used for speech recognition in particular. It is a method to reduce the number of dimensions for a given feature vector while features belonging to the same class are close together but features from different classes are far apart. As lots of the used features usually contain no relevant information reducing the feature dimension has been a sensible approach towards improving the performance of a recognition system.

Given a set of vectors \( x_i, i = 1, \ldots, M \) belonging to one of class \( \{ C_1, C_2, \ldots, C_c \} \), the within-class scatter matrix \( S_w \) (or covariance matrix) and the between-class scatter matrix \( S_b \) are defined as:

\[
S_b = \frac{1}{c} \sum_{k=1}^{c} (\mu_k - \mu)(\mu_k - \mu)^T
\]

(6.1)

\[
S_w = \frac{1}{M} \sum_{i=1}^{c} \sum_{x_k \in C_i} (x_k - \mu_i)(x_k - \mu_i)^T
\]

(6.2)

where \( \mu \) represents the total mean of all classes and \( \mu_i \) is the mean of class \( C_i \) (Class-specific mean vector). If the number of sample patterns belonging to class \( C_i \) is assumed as \( N \) the \( \mu_i \) can be as

\[
\mu_i = \frac{1}{N} \sum_{j=1}^{N} x_{ij}
\]

(6.3)

\( S_b \) indicates the deviation between the expected vectors for each pair of classes, while \( S_w \) shows the scatter of samples around the expected vector of their own class. It is desired to minimise \( \det(S_w) \) and maximise \( \det(S_b) \) simultaneously or

\[
Max \left| \frac{S_b}{S_w} \right| = Max |S_w^{-1}S_b|
\]

(6.4)
using techniques of linear algebra. The objective of LDA is to find a transformation
matrix $A$ to maximise the ratio:

$$\text{Max} \frac{|S_b|}{|S_w|} = \text{Max} \frac{A S_b A^T}{A S_w A^T}$$  \hspace{1cm} (6.5)$$

It can be shown that this is achieved by selecting the first greatest $m$ eigenvectors with
the discrimination matrix $S_w^{-1} S_b$, whose eigenvectors ($\phi_i, i = 1, 2, ..., n$) are ordered by
value of their eigenvalues $\lambda_1 > \lambda_2 > \lambda_3 > ... > \lambda_n$ and $A$ is as the following [86]:

$$A = [\phi_1, \phi_2, \phi_3, ..., \phi_m]$$  \hspace{1cm} (6.6)$$

The new feature vector is:

$$y = A(x - \mu)$$  \hspace{1cm} (6.7)$$

where $x$ is original feature vector, $y$ is the new feature vector and $\mu$ is the mean vector.

The scatter matrices for the new data samples, are then both diagonal which means
the coefficients are uncorrelated.

In the context of speech recognition, LDA is usually employed to reduce the feature vec­
tor dimensionality while retaining or even improving recognition performance. LDA has
been used successfully in speech recognition. It was first applied to speech recognition
by Hunt in independent Mel-scale linear discriminant analysis[25, 6]. The application
of LDA to improve the recognition accuracy of the two-dimensional cepstrum has been
investigated based on different types of distance measure in this paper [56].

### 6.3 Experiments with LDA

#### 6.3.1 TDC with LDA

The LDA transformation matrices are computed for TDC features. Based on these
LDA transformation matrices, the new feature matrices are computed. Table (6.1)
shows the results of applying the LDA to recognition accuracy for TDC method. By
comparing the two part of this table (with LDA and without LDA) it is observed that
the recognition accuracy is increased significantly (see table 6.3 statistical results) for
most classifiers and different dimensions of feature sub-matrices after applying LDA
method. As an example, the recognition accuracy for a sub-matrix of 20 × 3 and Euclidean classifier is 50.0 % before applying the LDA (table 5.9) while it is increased to 82.7 % after LDA. It means that recognition accuracy is increased 32.7 % in this case. According to the table (6.1) the largest increase in accuracy corresponds to the Euclidean classifier for all sizes of feature sub matrices. In contrast, there is no improvement in recognition accuracy for Mahalanobis classifier. Most probably the main reason for no improvement in the Mahalanobis distance is due to the fact that we used the total covariance of training data as a fixed matrix instead of an individual covariance matrix for each word. Because it is difficult to obtain accurate estimates for each covariance matrix (i.e \( C_i \)) from limited training data for each word, using one covariance matrix saves memory and increases the computational speed. Applying the LDA to the features cause the covariance matrix of each class to become close to the unit matrix not the total covariance matrix.

Table 6.1: The results of TDC with and without LDA

<table>
<thead>
<tr>
<th>Sub matrix</th>
<th>Feature Number</th>
<th>TDC with LDA</th>
<th>TDC without LDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>20×1</td>
<td>20</td>
<td>70.8</td>
<td>71.7</td>
</tr>
<tr>
<td>20×2</td>
<td>40</td>
<td>80.0</td>
<td>80.0</td>
</tr>
<tr>
<td>20×3</td>
<td>60</td>
<td>82.7</td>
<td>81.8</td>
</tr>
<tr>
<td>20×4</td>
<td>80</td>
<td>84.5</td>
<td>81.8</td>
</tr>
<tr>
<td>20×5</td>
<td>100</td>
<td>83.6</td>
<td>80.5</td>
</tr>
</tbody>
</table>

6.3.2 TDRC with LDA

The LDA transform matrices are computed, as for the TDC method, for the TDRC features and the new feature matrices are calculated. Table (6.2) shows the results. Two important issues can be observed by comparing this table to table (5.10) (before applying the LDA). Firstly, as expected, the LDA improved the recognition accuracy extensively for most classifiers and size of TDRC matrices. Secondly, applying the LDA to TDRC based on Euclidean classifier and Minkowski achieved the maximum
6.3. Experiments with LDA

recognition accuracy while there is no improvement in Mahalanobis classifier. For
instance, the recognition accuracy for a sub-matrix of $20 \times 3$ and Euclidean classifier is
62.7 % before applying the LDA (table (5.10)) while it is improved to 85.4 % after LDA.
Thus, there is about 22.7 % increase in recognition accuracy. Another observation can
be seen when comparing the result of TDC and TDRC after LDA i.e table (6.1) and
table (6.2). The maximum accuracy which the TDRC offers (table 6.2) is 85.4 % and
corresponds to 60 features and Euclidean classifier whereas it is 84.5 % for TDC and
corresponding to 80 feature number and the same classifier. This means that using the
TDRC not only increases the accuracy but also decreases the feature numbers.
Table 6.3 shows the results of a statistical analysis using the test described in [31]
for both TDC and TDRC with and without LDA. According to this table applying
the LDA gave a better performance which is statistically significant compared to the
without LDA for three of distance measures.
Furthermore, Fig.6.1 shows the results of recognition accuracy before and after LDA
for TDRC method as a function of $\gamma$ variation when the size of sub-matrix is $20 \times 3$. As
we emphasised before, the LDA improves the accuracy significantly for most of distance
classifiers (see statistical result) except for the Mahalanobis. According to this figure
the optimum value of $\gamma$ which maximises the recognition accuracy after applying the
LDA also is around 0.3 for all of them.

Table 6.2: The results of TDRC with LDA

<table>
<thead>
<tr>
<th>Sub Matrix</th>
<th>Feature Number</th>
<th>TDRC with LDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>20×1</td>
<td>20</td>
<td>73.6</td>
</tr>
<tr>
<td>20×2</td>
<td>40</td>
<td>81.5</td>
</tr>
<tr>
<td>20×3</td>
<td>60</td>
<td>85.4</td>
</tr>
<tr>
<td>20×4</td>
<td>80</td>
<td>83.6</td>
</tr>
<tr>
<td>20×5</td>
<td>100</td>
<td>82.7</td>
</tr>
</tbody>
</table>
Chapter 6. Developing the proposed methods

Table 6.3: Statistical analysis of comparison of TDC and TDRC with and without LDA

<table>
<thead>
<tr>
<th>Name of Methods</th>
<th>Statistical significant?</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDC with and without LDA at $20 \times 3$ sub-matrix</td>
<td>Yes</td>
</tr>
<tr>
<td>TDRC with and without LDA at $20 \times 3$ sub-matrix</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Figure 6.1: Results of recognition accuracy (TDRC method) before and after LDA
6.4 Mel-scale frequency TDRC

As stated in chapter 3, mel-frequency cepstral coefficients (MFCCs) are commonly used in many speech recognition systems. MFCCs are based on the known variation of the human ear's critical bandwidths with frequency. Mel-scale frequency is distributed linearly in the low frequency range but logarithmically in the high frequency range corresponding to the physiological characteristics of the human ear. Conventional mel-cepstrum features are one dimensional where a log function is applied in the process of computing the coefficients and widely used as suitable features to represent any frame of utterance. The novel contribution described here firstly extends this to a two dimensional cepstrum and secondly suggests a root function to replace the log function. Thus, to improve the recognition accuracy we generalise the TDRC by applying the Mel-frequency and modifying some stages in the procedure of the TDRC method. It is summarised as follows:

1. Preprocessing the speech including: Low pass filter, pre-emphasis, blocking to frames and Hamming window.

2. Compute the mel frequency root spectrum coefficients for each frame of speech. This stage includes spectral analysis, mel scale filter bank analysis and the application of the root function.

3. Applying the discrete Cosine transform on each frame to produce the mel frequency root cepstrum (MFRCC).

4. Gather the MFRCC of frames into a MFRCC matrix.

5. Take Inverse One-dimensional Fourier transform along the Horizontal direction in MFRCC matrix.

6. Take real part of coefficient to form the MTDRC matrix.

Therefore it can be concluded that the MTDRC features are derived from the following formula:

\[ c_{n,m} = \text{Real} \left\{ \sum_{k=1}^{M} \sum_{i=1}^{L} \beta[(E(i)]^2 \cos(n(0.5)i - 0.5)\pi/L)e^{2\pi km/M} \right\} \]
Chapter 6. Developing the proposed methods

\[ \text{for } n = 1, 2, \ldots, T \quad \text{and} \quad m = 1, 2, \ldots, M \] (6.8)

where \( T \leq N \) is the number of coefficients which are used per frame and \( M \) is number of frames, \( E(i) \) is the energy of the \( i^{th} \) filter, \( L \) is the number of filters in the desired bandwith and \( \beta = \sqrt{2/L} \). These new coefficients i.e MTDRC coefficients are represented in matrix form. Once again, analysis results show that the coefficients located at the lower index portion of the MTDRC matrix are more significant than elsewhere. Therefore, in the last stage a sub matrix is selected from the lower dimensions of the original matrix.

6.5 Experiments with Mel TDC and Mel TDRC

In this section the results of the mel scale TRDC are presented. To make a comparison the original TDC also has been modified by applying the mel scale frequency and result of mel TDC are illustrated here as well.

6.5.1 Experiments with MTDC

Table (6.4) shows the results of Mel TDC and original TDC after LDA. Comparing the two main part of the table (6.4) shows an improvement in accuracy when Mel scale is used. The maximum increase is related to Euclidean and Minkowski classifier. For instance the recognition accuracy for TDC method when feature sub-matrix is \( 20 \times 3 \) and Minkowsi classifier is 81.8 % whereas it is increased to 87.3 % for Mel TDRC under the same conditions.

6.5.2 Experiments with MTDRC

Table (6.5) shows the result of Mel TDRC which is suggested in this thesis. Firstly we compare the result of original TDRC (table (6.2) after LDA) and Mel TDRC. As can be expected, the recognition accuracy reaches 91.6 % for the Mel TDRC. The worst result of recognition have been obtained for Mansour classifier when the feature sub-matrix is \( 20 \times 3 \).

Secondly we compare the result of Mel TDC with Mel TDRC. Once again, table (6.4)
and (6.5) shows that the recognition accuracy of MTDC is 86.4 % for sub matrix of 20 × 3 and corresponding to Euclidean classifier while it is 91.6% in the MTDRC which shows about 5.2 % increase in accuracy.

Fig. 6.2 shows the result of recognition accuracy, where the solid curve and dashed line curve denote Mel TDRC and original TDRC respectively, as a function of γ variation when the size of sub-matrix is 20 × 3 and for different classifiers. As shown the Mel TDRC offers much better accuracy than the original TDRC for all classifiers. The best performance is achieved by the Euclidean classifier. The optimum value of γ which maximise the accuracy for Mel TDRC is around 0.2 while for the original TDRC is around 0.3. To sum up, as predicted, the mel scale improved accuracy for both of the TDC and TDRC significantly.

Table 6.5: The results of Mel TDRC

<table>
<thead>
<tr>
<th>Sub Matrix</th>
<th>Feature Number</th>
<th>Mel TDRC (MTDRC)</th>
<th>TDC (with LDA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20×1 20</td>
<td></td>
<td>82.7</td>
<td>80.9</td>
</tr>
<tr>
<td>20×2 40</td>
<td></td>
<td>87.3</td>
<td>87.3</td>
</tr>
<tr>
<td>20×3 60</td>
<td></td>
<td>86.4</td>
<td>87.3</td>
</tr>
<tr>
<td>20×4 80</td>
<td></td>
<td>89.1</td>
<td>89.1</td>
</tr>
<tr>
<td>20×5 100</td>
<td></td>
<td>86.3</td>
<td>85.5</td>
</tr>
</tbody>
</table>
6.6 Hartley Transform

The HT is a well known orthogonal transform similar to the FT [34]:

$$H(w) = \int_{-\infty}^{\infty} s(t)[\cos(wt) + \sin(wt)]dt$$  \hspace{1cm} (6.9)$$

The inverse Hartley is given by:

$$s(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} H(w)[\cos(wt) + \sin(wt)]dw$$  \hspace{1cm} (6.10)$$

It is closely related to the Fourier transform. By using the definition of Fourier transform of $s(t)$ it can be concluded that the relationship between the Hartley and Fourier transform is as follows:

$$H(w) = S_R(w) + S_I(w)$$  \hspace{1cm} (6.11)$$

where the $S_R(w)$ and $S_I(w)$ are the real and the imaginary parts of Fourier transform of $s(t)$ respectively. The HT has some advantages over the FT for the real signal. The first is that the HT of a real signal is a real spectrum. The second is that, as can be
seen from equations (6.9) and (6.10), its forward and inverse forms are identical thus simplifying implementation. Moreover it has the benefits of fast implementation at no loss of accuracy.

One of the other advantages of the HT which has not been considered in speech recognition previously, is that it retains the details of phase information of the spectrum in a single function, while in the FT, these are kept as two separate spectra, magnitude and phase. Though in the past, phase has been considered unimportant, recent research investigate that inclusion of phase in features can improve recognition accuracy [83, 4].

Let
\[ P(w) = |S(w)| = [S_r(w)^2 + S_i(w)^2]^{1/2} \]  \hspace{1cm} (6.12)
and
\[ Q(w) = |H(w)| = ([S_r(w) + S_i(w)]^2)^{1/2} \]  \hspace{1cm} (6.13)
where \( P(w) \) and \( Q(w) \) are the magnitude spectra of the FT and the HT respectively, \( S_r(w) \) and \( S_i(w) \) are the real and the imaginary spectral components common to both transforms. By expanding equation (6.13), it can be shown that:
\[ Q(w) = P(w) + S_r(w)S_i(w) + \ldots \text{(higher terms)} \]  \hspace{1cm} (6.14)

It is established, therefore, that \( Q(w) \) completely includes the Fourier magnitude spectral information, together with a convergent power series of \( S_r(w)S_i(w) \) terms, representing partial phase information. Thus, while in the case of the FT, the magnitude and phase information are represented in two separate and distinct spectral functions, in the case of the HT the absolute value of the HT spectrum contains magnitude information together with some partial phase information. Fig.6.3 shows a frame of a voiced speech signal and its FT and HT. As shown, the magnitude of the FT is an even function where only half contains information corresponding to the magnitude of signal while the phase information is kept in the other, odd, signal. The HT keeps both magnitude and phase in a single spectrum. Moreover the absolute value of the HT is a non symmetric function and keeps extra information corresponding to phase, as illustrated in equation (6.14).
Figure 6.3: The Fourier and the Hartley transform of a frame of voiced speech signal

The HT can be defined for a discrete signal $s(n)$ as the following:

$$H(\Omega) = N^{-1} \sum_{n=0}^{N-1} s(n)[\cos(\Omega n) + \sin(\Omega n)]$$  \hspace{1cm} (6.15)

and the inverse transform:

$$s(n) = \sum_{\Omega=0}^{N-1} H(\Omega)[\cos(\Omega n) + \sin(\Omega n)]$$  \hspace{1cm} (6.16)

More details about the properties of Hartley transform and Whitened Hartley Spectrum have been described in [16, 88].

6.6.1 Two-dimensional Hartley transform

There are two ways of defining the two-dimensional Hartley transform. The first one has an inseparable operator. The second one, which is adopted here, has a separable operator and can be defined as follows [117]:

$$H(f_1,f_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} s(t_1,t_2) \cos(2\pi f_1 t_1) \cos(2\pi f_2 t_2) dt_1 dt_2$$  \hspace{1cm} (6.17)

and the inverse is given by:

$$s(t_1,t_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} H(f_1,f_2) \cos(2\pi f_1 t_1) \cos(2\pi f_2 t_2) df_1 df_2$$  \hspace{1cm} (6.18)
where:

\[ \text{cas}(2\pi ft) = \cos(2\pi ft) + \sin(2\pi ft) \]  \hspace{1cm} (6.19)

\( H(f_1, f_2) \) is the two-dimensional HT of \( s(t_1, t_2) \). The similar definition for a two-dimensional discrete signal can be defined by replacing the integral with a double summation.

### 6.6.2 Hartley cepstrum

The HT has similar deconvolution properties to the FT and can be used as a substitute for the FT in the process of cepstrum analysis [15]. The benefits of using the HT is about 50 percent reduction in the data memory required and 40 percent decrease in program execution time when compared to the Fourier transform applied to cepstrum [108]. Here an alternative approach is proposed, as mentioned in [15], where the phase information is retained.

Consider the speech signal as the convolution of excitation sequence \( e(t) \) with the impulse response of the vocal tract filter \( v(t) \).

\[ s(t) = u(t) * e(t) \]  \hspace{1cm} (6.20)

Applying the HT to the Eq.(6.20) we have [88]:

\[ H_s(w) = \frac{1}{2}[H_v(w)H_e(w)+H_v(-w)H_e(w)+H_v(w)H_e(-w)-H_v(-w)H_e(-w)] \]  \hspace{1cm} (6.21)

where \( H_s(w) \), \( H_v(w) \), \( H_e(w) \) are the HT of \( s(t) \), \( v(t) \), \( e(t) \) respectively. By assuming that all phase information is associated with the excitation, the vocal tract impulse response \( v(t) \) may be assumed non-causal and an even function of \( w \). Thus, the HT of \( v(t) \), \( H_v(w) = H_v(-w) \) due to the even symmetry, under this condition; the Eq. (6.21) simplifies to Eq. (6.22).

\[ H_s(w) = H_v(w)H_e(w) \]  \hspace{1cm} (6.22)

This product can then be converted to addition by taking logarithms of the magnitudes of both sides of (6.22). In this last step, the inverse Hartley transform is taken to produce the Hartley cepstral coefficients \( c_h(\tau) \),

\[ c_h(\tau) = \mathcal{H}^{-1}[\log H_s(w)] = \mathcal{H}^{-1}[\log H_v(w)] + \mathcal{H}^{-1}[\log H_e(w)] \]  \hspace{1cm} (6.23)
where $H^{-1}$ indicates the inverse HT. There is one difficulty with equation (6.23) associated with the logarithmic function, while $H_s(w)$ is real, it also takes on both positive and negative values. However, the logarithm of a negative number may be defined as:

$$\log(x) = \log|x| + j\pi \quad \text{for } x < 0$$  \hspace{1cm} (6.24)

Thus:

$$\log[H_s(w)] = \log|H_s(w)| + j\pi b(w)$$  \hspace{1cm} (6.25)

where $b(w)$ is some binary sequence in the frequency domain. As a result, the HT is a complex function containing a real component and an imaginary part. However, the real component is associated with the modulus of the HT of the speech signal which contains both the main magnitude information and some partial phase information. The imaginary part is related to the transitions between positive and negative values of the Hartley spectrum, which also contains phase information [15].

An alternative approach which improves upon the capability of the Hartley cepstrum develops a Hartley cepstrum in two dimensions in order to preserve both static and dynamic features of the speech signal. Moreover, log function is replaced with a root function due to improved noise immunity of the root function [3]. Therefore, we introduce the two dimensional root Hartley cepstrum as described in the following section.

### 6.7 Hartley Two-dimensional Root Cepstrum

As in most of feature extraction methods including the two dimensional cepstrum, the magnitude of the FT is used and, therefore, information corresponding to phase is lost. To overcome this drawback, an alternative approach is proposed here, where features are extracted from a spectral representation but where both magnitude and partial phase details of the speech signal are included simultaneously. Here we suggest the application of the HT as a substitute for the FT in two-dimensional cepstrum analysis. The Hartley Two-dimensional Root Cepstrum (HTDRC) can be defined as the two-dimensional inverse Hartley transform of the root of the Hartley spectrum of a speech
6.7. Hartley Two-dimensional Root Cepstrum

signal.

\[
c(u, v) = \frac{1}{NM} \sum_{m=0}^{M-1} \sum_{k=0}^{N-1} H_\gamma(m, k) \cos(j2\pi kv/N) \cos(j2\pi mu/M)
\]

\text{for } 0 \leq u \leq N - 1, 0 \leq v \leq M - 1 \quad (6.26)

where the \( H_\gamma(m, k) \) is given by:

\[
H_\gamma(m, k) = H(m, k)\gamma \quad -1 \leq \gamma \leq +1 \quad (6.27)
\]

and \( H(m, k) \) is the HT of \( s(m, n) \), with \( n \) being the \( n^{th} \) sample in frame \( m \) of the speech signal, \( N \) is the number of samples in a frame, and \( M \) is the number of frames used for computing the HTDRC matrix. As with TDRC, the axis \( v \) is called quefrency and has time dimension. The axis \( u \) is called time frequency and has frequency dimension.

![Block diagram of Two-dimensional Hartley Root Cepstrum analysis](image)

Figure 6.4: Block diagram of Two-dimensional Hartley Root Cepstrum analysis

6.7.1 Experiments with HTDRC

Fig.6.4 shows the details of computing the TDHRC clearly. According to the separable definition of the two-dimensional HT the final stage of this figure can be replaced by
two one dimensional Hartley transforms. The first of them will be applied to the row of the feature matrix and the second will be applied to the columns. Table (6.6) shows the recognition accuracy using the HTDRC (involving Hartley transform) suggested in this chapter. Comparing this table and table (5.10)(original TDRC) shows that the HTDRC outperforms the TDRC for most feature matrices and all of classifiers. The recognition accuracy for HTDRC at a feature matrix of $20 \times 3$ is 94.5 % while it is 88.6 % for the TDRC. It shows improvement in recognition accuracy. Fig.6.5 shows the recognition accuracy as a function of $\gamma$ variation for TDRC, MTDRC and HTDRC. This figure indicates that the HTDRC offers much better results than either TDRC or MTDRC methods. The main reason is related to the phase information which is kept in the HT as explained before. According to this figure the optimum value for $\gamma$ which gives the maximum recognition accuracy is about 0.3 for TDRC, 0.2 for MTDRC and 0.4 for HTDRC.

Table 6.6: The results of HTDRC

<table>
<thead>
<tr>
<th>Sub Matrix</th>
<th>Feature Number</th>
<th>Hartley T D R C</th>
</tr>
</thead>
<tbody>
<tr>
<td>20x1</td>
<td>20</td>
<td>83.6</td>
</tr>
<tr>
<td>20x2</td>
<td>40</td>
<td>91.8</td>
</tr>
<tr>
<td>20x3</td>
<td>60</td>
<td>94.5</td>
</tr>
<tr>
<td>20x4</td>
<td>80</td>
<td>93.6</td>
</tr>
<tr>
<td>20x5</td>
<td>100</td>
<td>92.7</td>
</tr>
</tbody>
</table>

In conclusion, experimental results demonstrate the enhanced capability of the HT in two dimensional root cepstral analysis to improve recognition accuracy.

6.8 A comparison of feature extraction methods

In this section we report the results of the experiments performed in a comparison of different types of feature extraction approaches for speech recognition.
6.8. A comparison of feature extraction methods

Figure 6.5: Recognition accuracy as a function of $\gamma$ variation

6.8.1 Names of all used methods

These approaches are divided into two main categories: One dimensional cepstrum and Two dimensional cepstrum. The first group, one dimensional cepstrum, are listed as follows:

1. **ODC**: Conventional One-dimensional Cepstrum [14, 71].
2. **ODRC**: Conventional One-dimensional Root Cepstrum [3, 48].
3. **MODC**: Mel One-dimensional Cepstrum which is widely used in speech recognition systems [21].
4. **MODRC**: Mel One-dimensional Root Cepstrum.
   The second group, two dimensional cepstrum, also can be listed as follows:
5. **TDC**: Original Two dimensional cepstrum [5, 78].
6. **TDRC**: Two-dimensional root cepstrum, suggested in this thesis [17, 56, 59].
7. **MTDC**: Mel Two-dimensional cepstrum [45].
8. **MTDRC**: Mel Two-dimensional root cepstrum, suggested in this thesis.
6.8.2 Comparative criteria

To compare the methods two measures are used. First, it is necessary to test how good the features are. Thus, as a first measure we consider the separability criteria. It can be used to evaluate the scene discrimination capability of the selected feature from different types of feature extraction methods. If this factor is large enough, then the corresponding feature has good class separability. This means that the feature vectors which belong to the same class are close to each other while the feature vector from different classes are clearly separable and far from each other. This measurement of class separability is originally introduced as a criterion for the LDA and namely the trace criteria [28] and is defined as follows:

\[ J_{tr} = \text{tr}(S_w^{-1}S_b) \]  

(6.28)

where \( S_w \) denotes the within-class scatter matrix (or covariance matrix) and \( S_b \) the between-class scatter matrix as we mentioned in section (6.2). The notation \( \text{tr} \) is the trace of matrix which is the sum of the diagonal elements of a matrix and can be used to convert the matrix product \((S_w^{-1}S_b)\) to a scalar.

The second parameter for comparison, is the recognition accuracy which we have used before. In order to make a comparative study of feature extraction approaches meaningful it is essential that all methods use the same information. Thus in our study all methods employ an identical experiment set up. In computing the recognition accuracy, the LDA method has been applied to the features which have been extracted from the speech by all methods to achieve extra dimensionality reduction while preserving as much of the class discriminatory information and improving the recognition accuracy.

6.8.3 Experiments

In this section we describe the experiments carried out in order to compare the different types of methods. The aim of experiments is to compare the methods from the point of view of recognition performance and a separability criterion.
6.8. A comparison of feature extraction methods

**Results of separability criterion**

Let's start with the results of a separability criterion. Fig.6.6 shows the Separability Criterion, $J$, as a function of feature number for different types of one-dimensional and two-dimensional feature extraction methods. These methods have been described previously in the second section of this chapter. Some important issues can be observed from this figure. Firstly, for all methods, the separability is increased when feature number is increased. Secondly, the 2-D methods i.e HTDRC, MTDRC and TDC provide larger separability than 1-D methods i.e MODC and ODC method. Thirdly, the best performance is achieved by HTDRC. This means that the features which are produced by this method are more separable than others and makes the classification stage easier. Moreover to have a total idea about the separability for all methods, Fig.6.7 show the maximum amount of separability for all 1-D and 2-D methods when number of features are fixed. Comparing the two parts of this figure show that once again all 2-D methods have larger separability criteria. Among the 1-D methods the MODRC has the highest separability and ODC has the lowest. Among the 2-D methods the highest separability is achieved by THDRC while the lowest is offered by the original TDC. As can be seen from the Fig.6.7.b MTDRC has the second largest separability and after that the TDRC provides better separation.

**Results of recognition accuracy**

Fig.6.8.a shows recognition accuracy for different types of 1-D methods when the total feature number is 100, while Fig.6.8.b shows the recognition accuracy for different types of 2-D method when the total feature number is 60. According to the Fig.6.8 recognition accuracy of all 2-D methods is larger than 1-D methods. Furthermore, the total number of features which are used in 2-D methods are 60 which are much less than 100 in 1-D methods. Thus, this reduces computation and increases the speed. Among the 1-D methods (Fig.6.8.a) the MODRC offers the best accuracy while the ODC offers the lowest accuracy. It is noticed that by referring to Fig.6.7.a we can see that the MORDC also offers the largest separability in this group. Among the 2-D methods the highest accuracy is achieved by HTDRC while the lowest is offered by
Chapter 6. Developing the proposed methods

The aim of this chapter was to develop the proposed feature extraction methods described in chapter 4 and 5 using different techniques. The LDA was applied to the methods at the first stage to improve the results. Applying the LDA causes an improvement in the accuracy for both TDC and TDRC and most of classifiers. In this chapter we introduced the Mel TDRC as a modified method to improve the accuracy of TDRC. As in most feature extraction methods, including the two dimensional cepstrum, the magnitude of the FT is used, information corresponding to phase is lost. In the next step, the HT was suggested as a substitute for the FT in two-dimensional cepstrum analysis to preserve both magnitude and phase details of the speech signal.

Figure 6.6: Separability Criteria using different types of features extraction methods

As can be seen from the Fig.6.8.b MTDRC has the second largest accuracy and after that the TDRC provides better accuracy. Moreover table (6.7) shows more details of recognition accuracy for all types of 1-D and 2-D features. This tables emphasises again that the 2-D methods offer much better accuracy than 1-D methods.

6.9 Summary and conclusion

The aim of this chapter was to develop the proposed feature extraction methods described in chapter 4 and 5 using different techniques. The LDA was applied to the methods at the first stage to improve the results. Applying the LDA causes an improvement in the accuracy for both TDC and TDRC and most of classifiers. In this chapter we introduced the Mel TDRC as a modified method to improve the accuracy of TDRC. As in most feature extraction methods, including the two dimensional cepstrum, the magnitude of the FT is used, information corresponding to phase is lost. In the next step, the HT was suggested as a substitute for the FT in two-dimensional cepstrum analysis to preserve both magnitude and phase details of the speech signal.
Table 6.7: Recognition accuracy (%) for all types of methods

<table>
<thead>
<tr>
<th>Feature Number</th>
<th>One-D methods</th>
<th>Two-D methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>ODC</td>
<td>MODC</td>
</tr>
<tr>
<td>20</td>
<td>56.4</td>
<td>52.7</td>
</tr>
<tr>
<td>40</td>
<td>58.9</td>
<td>60.0</td>
</tr>
<tr>
<td>60</td>
<td>64.2</td>
<td>66.4</td>
</tr>
<tr>
<td>80</td>
<td>70.0</td>
<td>71.3</td>
</tr>
<tr>
<td>100</td>
<td>72.7</td>
<td>77.3</td>
</tr>
</tbody>
</table>

Separability Criteria for One dimensional methods

Separability Criteria for Two dimensional methods

(a) One-dimensional methods
(b) Two-dimensional methods

Figure 6.7: Separability Criteria of one and Two-dimensional methods
Chapter 6. Developing the proposed methods

Figure 6.8: Recognition accuracy for One-dimensional and Two-dimensional methods

simultaneously. Experimental results demonstrate the enhanced capability of the HT in two dimensional root cepstral analysis to improve recognition accuracy.

At the end of this chapter, an experimental comparative study of 9 kinds (various approaches) of feature extraction methods for isolated word speech recognition were also carried out. The methods investigated include two main categories, conventional one-dimensional and two-dimensional methods. The experiments were conducted on TIMIT data base. The results of experiments demonstrated the superiority of the two dimensional cepstrum method over the all kinds of one dimensional method in terms of both recognition accuracy and separability criteria. Most probably the success of two-D methods is owed to the fact that representation of both the instantaneous and transitional information of a speech signal is included by using the 2-D methods, as discussed in chapter 4. Among the different types of the 2-D methods the Hartley TDRC and Mel TDRC which are suggested in this thesis are the best.
Chapter 7

Conclusion and future work

7.1 Summary of the thesis

This thesis focuses on feature extraction for speech recognition. In chapter 3 an overview of different conventional approaches which are used for feature extraction methods in current speech recognition system have been described. Among different types of feature extraction methods, which are mentioned in chapter 3, MFCC and PLP are the most popular. Both methods employ an auditory-based warping of the frequency axis, derived from the frequency sensitivity of human hearing. MFCCs are based on a uniform spacing along the mel-scale whereas PLP uses the Bark scale. However there are significant differences in the process of computing the PLP coefficients and MFCC coefficients. The main difference between PLP and MFCC is related to the output cepstral coefficients. The PLP model uses an all pole model to smooth the modified power spectrum. In contrast, Mel scale cepstral analysis uses cepstral smoothing to smooth the modified power spectrum. PLP provides a more robust representation than MFCC yet in spite of this, MFCC are historically more popular probably because MFCC is simpler to implement [65].

Most of conventional feature extraction methods imply one dimensional, either time or frequency, and omit important features for recognition which exist in both time and frequency. Moreover, to include the speech transitional dynamic, it is necessary to apply further steps to obtain more information (dynamic information) from the speech by
using delta or delta delta features as discussed in chapter 3. These methods normally need a time warping technique, while an acoustic image (or two dimensional features) is an alternative approach which has some advantage over the one dimensional methods. Perhaps the most familiar of two dimensional features is the spectrogram however other forms such as the two-dimensional cepstrum have more useful characteristics in speech recognition. Different types of two dimensional cepstrum has been discussed in chapter 4. It has been proposed that speech characteristics are better represented by an acoustic image, a two dimensional feature representation with time along one axis and frequency along the other. By using the two-dimensional features such as the TDC, the dynamic features and static features are represented simultaneously. The TDC separates the envelope and excitation signals from each other and has the ability of data compression which make the classification process easier. It represents the features of the speech signal in a matrix form. Analysis results show that the coefficients located at the lower index portion of the TDC matrix seem to be more significant than others. Hence to represent an utterance only some TDC coefficients need to be selected to form a feature matrix. Unlike the one-dimensional methods, in two-dimensional methods, a feature matrix is considered for a whole word instead of using a stream of vectors and avoids the need for DTW. Due to the use of the distance measure between the test feature matrix and reference, the recognition time is short.

We introduced the TDRC as a new feature extraction method to overcome the weakness of the original TDC especially in noisy conditions. In addition to the feature compression, which is more than TDC, the TDRC has some advantages over the TDC. It has the advantage of an adjustable $\gamma$ parameter which can be used to optimise the feature extraction process, reducing the dimensions of the feature matrix and giving simple computation. Although some primary experiments on the performance of the two dimensional cepstrum are given in chapter 4, the performance of the suggested methods are presented in chapter 5. It is shown that both the TDC and the TDRC outperform the conventional cepstrum. In this chapter, several effects of changing the TDC parameters such as changing the row number of TDC feature matrix, column number of TDC feature matrix or optimum frame length on recognition accuracy have been considered. Furthermore, the superior performance of TDRC over the TDC, es-
7.2 Future work

especially in noisy conditions is demonstrated in this chapter and the effects of TDRC feature matrix has been described. Some experiments have been carried out to investigate the optimum value of $\gamma$. According to our experiments, the optimum value for $\gamma$ is varied depending on the feature sub matrix, the type of distance classifier and the data. However this optimum value is around 0.3 for overall. Moreover some experiments have been performed to investigate the effects of different distance classifiers on the performance of speech recognition.

In chapter 6 the proposed feature extraction methods described in chapter 4 and 5 are developed using different techniques. Applying the LDA causes an improvement in the accuracy for both TDC and TDRC and most classifiers. In this chapter we introduced the Mel TDRC as a modified method to improve the accuracy of TDRC. As in most feature extraction methods, including the two dimensional cepstrum, the magnitude of the FT is used, information corresponding to phase is lost. In the next step, the HT was suggested as a substitute for the FT in two-dimensional cepstrum analysis to preserve both magnitude and phase details of the speech signal simultaneously. Experimental results demonstrate the enhanced capability of the HT in the two dimensional root cepstral analysis to improve recognition accuracy. An experimental comparative study of 9 kinds (various approaches) of feature extraction methods for isolated word speech recognition are also carried out in this chapter. The methods investigated include two main categories, conventional one-dimensional and two-dimensional methods. The experiments were conducted on TIMIT data base.

In summary, the results and findings of this thesis demonstrated the superiority of the two dimensional cepstrum method especial all types of the TDRC over the all kinds of one dimensional method in term of both recognition accuracy and separability criteria.

7.2 Future work

The presented work in this thesis can be extended in different ways. In this section we illustrate some possible extensions in the following directions.
Applying the DTW to two-dimensional features

As we know, two utterances of the same word can show large differences in the duration. Here, the variation in the length of the utterances used, has been taken care of, by appending zeros as in [77], where necessary. A fixed number of frames, depending on utterance length, is taken to form the 2-D matrix (all types of TDC, TDRC, MTDC,...). It is hoped that in doing so the durational variations are absorbed into the feature matrix.

In fact we used a kind of linear time alignment. However, DTW is the commonly used technique to address this problem. It is expected that applying the DTW could improve the results. For future work, applying the DTW to solve this problem and comparing this to the methods described here, would be of interest.

Density estimation

As we mentioned in chapter 5, we assumed that the pdf of the features obtained via a two dimensional cepstrum form a single Gaussian distribution. Also some primary experiments were carried out which assumed the GMM for the features. Therefore, one possible route for future work is to estimate the density functions of all TDC methods including the original TDC, Mel TDC, original TDRC, MelTDRC and Hartley TDRC. The suggested model for the features is a GMM. One of the most important problems to be considered, related to GMM, is to find an optimum number of Gaussian functions which gives the best performance.

Application to continuous speech recognition

Our suggested methods in this thesis were for isolated word speech recognition. Other possible future work is applying these methods to continuous speech recognition.

Application to speaker recognition

As features used for speech and speaker recognition systems are mostly common, it would be interesting to use the two dimensional features suggested here for a proposed
7.2. Future work

of speaker recognition system.

Audio classification

Audio classification is an important area of pattern recognition which employs almost the same feature extraction methods as speech. Another area for further work can be therefore the issue of applying two-dimensional cepstrum to the audio classification.

Improving the classifiers

In this research we used Bayse's rule for classification which can be simplified to Mahalanobis distance classifier under some assumptions. However, it would be interesting to use other kinds of classification such as ANN.
Appendix A

Details of Experimental Methodology

In this appendix the details of the experimental methodology are illustrated. In the first section, the general experimental set up is explained. In section two the training process is given, the testing process is presented in section three, and finally the recognition process is given in section 4.

A.1 General experimental set up

For the experiments described in the thesis, in total, we used 539 words selected from the sentences of the TIMIT database. The data was divided into test and training utterances as described below. From these words, 418 words, uttered by 38 speakers, were used in training while 121 words, uttered by 11 different speakers, were used for testing recognition performance. These words selected were chosen because these were common to all speakers which limited the size of the database.

For each recorded sentence in the TIMIT database there is a file which contains a record of corresponding beginnings and ends of each word in a sentence specified by sample position. We used these files to find the boundaries of each word. The details of these selected words are shown in table A.1. We considered a fixed number of frames for each
word depending on its length which should be an integer power of two as appropriate for the FFT computation.

Table A.1: List of words

<table>
<thead>
<tr>
<th>Word</th>
<th>Number in training</th>
<th>Number for test</th>
</tr>
</thead>
<tbody>
<tr>
<td>She</td>
<td>38</td>
<td>11</td>
</tr>
<tr>
<td>Had</td>
<td>38</td>
<td>11</td>
</tr>
<tr>
<td>Your</td>
<td>38</td>
<td>11</td>
</tr>
<tr>
<td>Dark</td>
<td>38</td>
<td>11</td>
</tr>
<tr>
<td>Suit</td>
<td>38</td>
<td>11</td>
</tr>
<tr>
<td>In</td>
<td>38</td>
<td>11</td>
</tr>
<tr>
<td>Greasy</td>
<td>38</td>
<td>11</td>
</tr>
<tr>
<td>Wash</td>
<td>38</td>
<td>11</td>
</tr>
<tr>
<td>Water</td>
<td>38</td>
<td>11</td>
</tr>
<tr>
<td>All</td>
<td>38</td>
<td>11</td>
</tr>
<tr>
<td>Year</td>
<td>38</td>
<td>11</td>
</tr>
<tr>
<td>Total</td>
<td>418</td>
<td>121</td>
</tr>
</tbody>
</table>

A.2 Training process

We applied the proposed feature extraction methods to every utterance in the training database and a feature matrix was produced for each utterance. Since, significant values of the features are located in the low order part of the feature matrix, from each feature matrix, one compact sub-matrix was extracted. Then, an average feature matrix was obtained for each of the classes. The average feature matrix of each class was obtained by taking the means of the corresponding elements for the set of contributing matrices. Thus each class is represented by only one compact matrix which is converted to a vector and then used as a reference vector in the classifier.

It should be mentioned that for the case where we assumed a GMM for the features
A.3 Testing process

Once again in the testing process, we applied the proposed feature extraction methods to each utterance in the test data. A compact feature matrix was produced for each utterance and then converted to the feature vector.

A.4 Recognition process

During the recognition process a test pattern is compared to all the reference patterns and a distance measure is calculated for each comparison. The decision rule is to find the reference pattern which gives the minimum distance.

In the case of the GMM (section 5.9) for a given feature test vector, the recognition process is to find the class which gives the highest probability (Maximum likelihood approach).

Since the speakers used in training and testing were different we consider these tests to be speaker independent.

The results have been produced by averaging over a different number of speakers tested on different occasions. This means that on the first occasion, all words uttered by one speaker have been tested, then on the second occasion, all words uttered by two different speakers from test data have been tested and so on. In the last stage all words uttered by all speakers of the test data have been tested. During each experiment the order of the utterances was maintained e.g in the experiment one, the words 'she', 'had', 'your' etc spoken by speaker one were presented in that order. In experiment two, the two words 'she', 'she', 'had', 'had', 'your', 'your' etc. were presented. The first word of each pair being speaker one and the second word being speaker 2. This procedure was repeated for three, four, up to the last speaker. An overall performance measure was found from these results for each experiment and recorded as a percentage accuracy.
Appendix B

Statistical significance test for comparing two classifiers

As mentioned in [31], it is important to investigate whether there is any significant difference between two algorithms in a speech recognition performance measurement. There are two important tests for this issue which have been described fully in [31].

A simple approach, appropriate to the data used here, can be considered as follows: for given algorithms $A_1$ and $A_2$ with the same set of $N$ test examples, the goal is to decide if one is superior to the other. We consider null hypothesis $H_0$ that the two algorithm have the same error rates and we seek whether there is evidence to reject $H_0$. If $A_1$ makes $n_1$ errors and $A_2$ makes $n_2$, the maximum likelihood estimates of the errors are given by:

$$\hat{e}_1 = \frac{n_1}{N}, \text{ for } A_1$$

(B.1)

and

$$\hat{e}_2 = \frac{n_2}{N}, \text{ for } A_2$$

(B.2)

The variances of $\hat{e}_1$ and $\hat{e}_2$, $\sigma_1^2$ and $\sigma_2^2$, also can be computed as follows:

$$\sigma_1^2 = \frac{\hat{e}_1 (1 - \hat{e}_1)}{N}$$

(B.3)

$$\sigma_2^2 = \frac{\hat{e}_2 (1 - \hat{e}_2)}{N}$$

(B.4)
Appendix B. Statistical significance test for comparing two classifiers

Then the difference between two errors-rates is given by

\[ d = |\tilde{e}_1 - \tilde{e}_2| \quad (B.5) \]

and the associated variance, \( \sigma_d^2 \), is:

\[ \sigma_d^2 = \text{Var}(\tilde{e}_1 - \tilde{e}_2) \quad (B.6) \]

Under the assumption that \( A_1 \) and \( A_2 \) are independent, equation B.6 can be written as:

\[ \sigma_d^2 = \text{Var}(\tilde{e}_1) + \text{Var}(\tilde{e}_2) = \sigma_1^2 + \sigma_2^2 \quad (B.7) \]

Therefore under this assumption the standard deviation of \( d \) can be computed as follows:

\[ \sigma_d = \sqrt{\frac{\tilde{e}_1(1 - \tilde{e}_1)}{N} + \frac{\tilde{e}_2(1 - \tilde{e}_2)}{N}} \quad (B.8) \]

By assuming \( N \) is large, the distributions of \( d \) can be considered as normally distributed. 95% of the values of the random variable \( d \) are included in the range of \( d \pm 2\sigma_d \). We define

\[ p = d - 2\sigma_d \quad (B.9) \]

and when \( p > 0 \), we can say that \( H_0 \) is rejected. In other words, with 95% confidence, there is a significant difference between \( \tilde{e}_1 \) and \( \tilde{e}_2 \) and only with a 5% of probability the performance of \( e_1 \) and \( e_2 \) are actually the same.

As an example, to illustrate this test, let us consider the effect of adding columns to the feature matrix on recognition accuracy (table (5.5)). We assume \( A_1 \) be the case when we have only one column in the feature matrix and \( A_2 \) be the case when we have three columns in the feature matrix. According to the experimental results, \( \tilde{e}_1 = 0.2640 \) and \( \tilde{e}_2 = 0.1440 \) thus:

\[ d = |\tilde{e}_1 - \tilde{e}_2| = 0.12 \quad (B.10) \]

From equation B.8 and by considering \( N = 121 \):

\[ \sigma_d = 0.0589 \quad (B.11) \]

Hence, \( p = d - 2\sigma_d = 0.0021 > 0 \) we can state that with 95% confidence there is significant difference between \( A_1 \) and \( A_2 \). In other words this test show that there is
significant increase in recognition accuracy when the coefficients of the third column are incorporated in the feature sub-matrix.

As a second example let us consider $A_1$ as the same of the first example but $A_2$ be the case when two columns are included. Here, we have $\hat{e}_1 = 0.2640$ and $\hat{e}_2 = 0.164$ thus

$$\tilde{d} = |\hat{e}_1 - \hat{e}_2| = 0.10$$

(B.12)

and once again using equation B.8:

$$\sigma_d = 0.086$$

(B.13)

Hence, $p = \tilde{d} - 2\sigma_d = -0.017 < 0$ there is no significant difference between $A_1$ and $A_2$.

As mentioned before, this simple approach only can be used under assumption that the algorithms make independent errors, more details can be found in [31].
Appendix B. Statistical significance test for comparing two classifiers
Bibliography


