Spring 2006

Whole Word Phonetic Displays for Speech Articulation Training

Fansheng Meng
Old Dominion University

Whole Word Phonetic Displays for Speech Articulation Training

by

Fansheng Meng

B. S. University of Electronic Science & Technology of China (UESTC)
M. S. University of Electronic Science & Technology of China (UESTC)

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirement for the Degree of

DOCTOR OF PHILOSOPHY

ELECTRICAL AND COMPUTER ENGINEERING

OLD DOMINION UNIVERSITY

May 2006

Approved by:

Dr. Stephen A. Zahorian (Director)

Dr. K. Vijayan Asari (Member)

Dr. Min Song (Member)

Dr. Shunichi Toida (Member)

ABSTRACT

Whole Word Phonetic Displays for Speech Articulation Training

Fansheng Meng
Old Dominion University, 2006
Director: Dr. Stephen A. Zahorian

The main objective of this dissertation is to investigate and develop speech recognition technologies for speech training for people with hearing impairments. During the course of this work, a computer aided speech training system for articulation speech training was also designed and implemented. The speech training system places emphasis on displays to improve children's pronunciation of isolated Consonant-Vowel-Consonant (CVC) words, with displays at both the phonetic level and whole word level. This dissertation presents two hybrid methods for combining Hidden Markov Models (HMMs) and Neural Networks (NNs) for speech recognition. The first method uses NN outputs as posterior probability estimators for HMMs. The second method uses NNs to transform the original speech features to normalized features with reduced correlation. Based on experimental testing, both of the hybrid methods give higher accuracy than standard HMM methods. The second method, using the NN to create normalized features, outperforms the first method in terms of accuracy. Several graphical displays were developed to provide real time visual feedback to users, to help them to improve and correct their pronunciations.

To my lovely parents and my family!

ACKNOWLEDGMENTS

I am immensely grateful to my advisor, Dr. Stephen A. Zahorian, for his valuable academic guidance and encouragement throughout the course of this study. His insight and wisdom greatly inspired me. To a great extent, my work ethic has been shaped by his professionalism, which has been an invaluable example to me. I would also like to thank my research committee members, Dr. K. Vijayan Asari, Dr. Min Song and Dr. Shunichi Toida for their review and valuable comments.

I am indebted to my parents and would like to thank them for their courage and support.

Special thanks go to my wife, Dr. Qin Hu for her understanding and valuable support when I stayed glued to my computer.

Special thanks also to Mrs. Penny Hix, Mr. Mukund Devarajan and Mr. Eugen Rodel for their useful suggestions and assistance.

Also, I would like to thank my colleagues in the Speech Communication lab for their help.

This work was partially supported by NSF grant BES-9977260.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1. Overview of Automatic Speech Recognition

Automatic Speech Recognition (ASR) consists of signal processing and pattern recognition used to automatically transcribe an acoustic speech signal into a sequence of words. More generally, it is the first critical step needed to understand the message carried in the speech.

ASR technology, along with human language technology, has been studied and developed for more than fifty years. Much progress has been achieved in this field [1]. For restricted vocabularies, ASR systems can achieve high performance, even under somewhat noisy conditions such as over the telephone. For very large vocabulary speaker-independent continuous speech recognition tasks, recognition performance in the laboratory is also impressive, but generally not so good under noisy conditions. In general, ASR can be used in the following fields:

1. Dictation: ASR can translate spoken words into written text. Applications include inputting words into a computer with speech instead of typing.
2. Command and Control: In these applications, users can control the computer and software applications by uttering predefined commands.
3. Embedded Applications: ASR has been applied to many embedded systems. These systems allow users to speak commands instead of pressing buttons to create specific actions.
4. Medical/Disabilities: Many people have difficulty typing due to physical limitations such as repetitive strain injuries, muscular dystrophy, and many others.

For example, people who have difficulty hearing could use a system connected to their telephone to convert the caller's speech to text.

5. Education: Some systems based on ASR technology help people to learn a new language.

A speech recognition system can be used in many different modes: speaker-dependent or independent, for isolated words or continuous speech, for small, medium or large vocabularies.

1. Speaker Dependent / Independent system:

A speaker-dependent system is one where the system is trained for a specific speaker. The speaker is usually asked to record predefined words or sentences that will be analyzed and whose analysis results will be stored by the system. In contrast, speaker-independent systems are not limited to a specified speaker, and the system can be used by any speaker without any training procedure. The speaker-dependent systems can be adapted to specified users and therefore the accuracy for the speaker-dependent mode is better compared to that of the speaker-independent mode.

2. Isolated Word Recognition / Continuous Speech Recognition

In isolated word recognition systems, each word is surrounded by silence so that word boundaries are straightforward to locate. The system does not need to solve the difficult problem of finding word boundaries in a sentence. Continuous speech recognition is much more natural. It assumes that the computer is able to recognize a sequence of words in a sentence. However, this mode requires much more CPU and memory, and the recognition accuracy is typically much lower than for isolated word recognition.

3. Vocabulary Size

It is clear that the larger the vocabulary is, the more opportunities the system will have to make errors. Therefore, one strategy for a good speech recognition system is to adapt its vocabulary to the task it is currently assigned to (i.e., possibly enable a dynamic adaptation of its vocabulary). The difficulty level can be rated with a score from 1 to 10 0, where 1 is the simplest system (speaker-dependent, able to recognize isolated words in a small vocabulary (10 words) and 10 corresponds to the most difficult task (speaker-independent continuous speech over a large vocabulary (say, 10,000 words). State-of-the-art speech recognition systems with acceptable error rates are somewhere between these two extreme difficulty levels.

Although ASR has made vast improvements over the past few decades and is usable for some applications, this technology is not a solved problem in any fundamental sense [3]. It remains a very hard problem. In some cases, only the ASR systems for very limited vocabulary size achieve acceptable performance. For continuous speech recognition tasks, there is no any system that works as well as human beings. The latter can recognize speech correctly for different people with different accents and even in noisy environments.

The reasons for the difficulties of speech recognition can be summarized as follows. First, the basic units of speech are hard to recognize. The phonemes, which are the smallest sound units used in most current speech recognition systems, are highly dependent on the context in which they appear. These phonetic variabilities make the pose great difficulties for ASR systems to be able to recognize the basic speech unit with simple rules. Second, continuous speech adds more to the difficulties encountered. The

acoustic signal for an identical word at different positions in a continuous sentence may vary greatly. Third, within-speaker variabilities can result from changes in the speaker's physical and emotional state, speaking rate, or voice quality. Fourth, acoustic variabilities can result from changes in the environment as well as in the position and characteristics of the transducer. Finally, the human language understanding process is still poorly understood.

To improve speech recognition performance, researchers are working in the following fields [4] [5] [6]:

1. Developing robust speech recognition algorithms. The algorithms include the optimization of speech models, search algorithms and applying new and robust pattern recognition algorithms for speech recognition tasks.
2. Noise and speaker adaptation. This area includes development of embedded systems for ASR that can recognize speech for different people and in different environments with higher accuracy.
3. Grammar and language models. For some particular tasks, the recognizers are able to achieve good performance with appropriate grammar and language models.

1.2 Speech training with ASR for hearing impaired people

The use of computers to give visual feedback to speakers with hearing impairments has been the subject of research for many years [7] [8]. The aim of ASR research in this field has been to make speech therapy more effective, not only by providing objective feedback to supplement the efforts of the therapist, but also by freeing the therapist from time-consuming practice sessions, and giving the hearing-

impaired users greater opportunities for practice using stimulating and motivational material.

Hearing impaired people usually have difficultly learning to produce intelligible speech. The key impediment is that they cannot hear their own speech, so that they do not receive the real time feedback that normal-hearing listeners receive whenever they speak. They do not know whether their pronunciation is correct unless others tell them using a non-auditory method. To help them practice their pronunciation, some devices [9] [10] have been developed to give them feedback by automatic methods. However, two primary reasons limit their wide spread use. One is their low recognition rate, and another is their expensive price. As ASR technology becomes more accurate, the use of ASR as a compensatory strategy for individuals with hearing impairments becomes more feasible.

A good speech training system should be a personal tutor to a user with unlimited time to spend. To achieve this objective, abundant and accurate information should be provided to the user. This requires that the system have a "strong" engine to process the speech signal. The system should extract useful speech message information from the original input acoustic signal and analyze the speech information and give users the results.

Adequate and motivational feedback is another important aspect for speech training. The system should give users "goodness" scores: users should be able to judge exactly which parts of their utterance are unacceptable, and be able to interpret the displays in a way which allows them to modify their production to make it more similar

to the intended target. Even more importantly, the users should be able to use the system without constant supervision by a therapist.

1.3 Scope of dissertation

The objective of this dissertation is to model acoustic-phonetic information for use in ASR and to create speech training aids for people with hearing impairments. Particular emphasis will be placed on how to improve children's pronunciation to an isolated Consonant-Vowel-Consonant (CVC) words at both the phone level and word level.

Speech recognition accuracy plays an important role in speech training. Hidden Markov Models (HMMs) are the most popular models for state-of-the-art ASR systems. This is because the HMM with its related algorithms is a powerful tool with high capability for modeling complex non-stationary phenomena. In this dissertation, methods for combining Hidden Markov Models with Neural Networks (HMM/NN) were investigated. The idea of hybrid systems comes from the idea that HMM and NN models can overcome each other's limitations and take advantage of their individual strength.

For speech training, another important consideration is how to visually present indicators of pronunciation "correctness" at the phone level in real time, with minimum time delays, in response to short words produced by a speaker. Several displays were developed to present user's speech.

The specific objectives of this work were to:

1. Investigate real time visual speech displays which can respond to the user's speech with minimum time delay.

2. To develop combination HMM/NN phonetic models and associated visual displays which can be used to assess the pronunciation quality of short words
3. To develop and make available a suite of software tools in phonetic recognition for use in speech training, so that the users can easily and quickly correct their pronunciation.

1.4 Outline of dissertation

Chapter 2 of this dissertation provides a brief description of signal processing for ASR, and then reviews some current speech training systems, including a vowel aid training speech system developed in the speech lab at ODU. Chapter 3 introduces hidden Markov models (HMMs) and shows some experimental results based on HMMs with a CVC database. In this method, the observation probabilities are estimated with Gaussian Mixture Models (GMMs). Chapter 4 and 5 describe two hybrid methods for speech recognition. In chapter 4, the neural networks work as the posterior probability estimator for HMMs. In chapter 5, the speech features are normalized with neural networks, and then HMMs are trained and tested with the normalized features. These two hybrid methods are shown to outperform the more typical HMM/GMM method. In Chapter 6, the details about the computer display for CVC training are discussed. Finally, in Chapter 7, conclusions and suggestions for future work are presented.

CHAPTER II

BACKGROUND

2.1 Signal processing for ASR

An ASR system usually consists of three major modules [11]: a speech signal processing front-end, acoustic modeling, and language modeling. The speech signal processing front-end is the first step of ASR for extraction of acoustically-invariant features from the original speech waveform. This is because much of the detail of the speech signal is not relevant or only marginally relevant to the speech message information. Usually, speech feature extraction is based on short-time spectral magnitude computations. The reason for that speech information has been shown to depend mainly on short-time spectral characteristics, primarily independent of overall gain and phase. The most commonly used speech feature extraction methods include:

2.1.1. Linear Predictive Coding (LPC) Modeling

Linear Predictive (LP) analysis [12] of speech is one of the most important speech analysis techniques. The basis of this technique is to use an all-pole filter to model the vocal tract. In other words, the vocal tract is considered as a filter and the goal is to determine the transfor function of the vocal tract. The signal can be represented as:

$$\hat{s}_n = \sum_{i=1}^{p} a_i s_{n-i} \tag{2.1}$$

where p is the number of poles. This linear filter has the transfer function:

$$H(z) = \frac{1}{1 - \sum_{i=1}^{p} a_i z^{-i}} \tag{2.2}$$

where the a_i are filter coefficients that are selected to minimize the overall mean squared error over the analysis frame.

There are three ways to determine LP coefficients: the covariance method, the harmonic method and the autocorrelation method. The autocorrelation method is mostly used in speech recognition because it yields stable filters and is very computationally efficient.

2.1.2 Bank-of-Filters Front-End Model

This model was designed to imitate the perceptual characteristics of the human ear which resolves frequencies non-linearly across the audio spectrum. It uses a set of band-pass filters whose center frequencies spead nonlinearly across the frequency range of speech. To implement the filterbank, a window of speech data is transformed using a Fourier transform and the magnitude is assessed. Usually, the magnitude coefficients are *binned* by correlating them with triangular bandpass filters. Thus, each bin holds a weighted sum representing the spectral magnitude in that filterbank channel.

There are two well-known perceptual frequency scales, the Bark scale and the Mel scale, used to determine filter spacing. The most popular scale is the Mel scale, and the features are called Mel-Frequency Cepstral Coefficients (MFCCs) [13]. The coefficients are calculated from the log filterbank amplitudes using the Discrete Cosine Transform:

$$c_i = \sqrt{\frac{2}{N}} \sum_{j=1}^{N} m_j \cos\left(\frac{\pi i}{N}(j-0.5)\right) \tag{2.3}$$

where m_j is log filterbank amplitudes and N is the number of filterbank channels.

The MFCCs introduced above are basic static parameters. In practice, the performance of a speech recognition system can be greatly enhanced by adding an energy coefficient and time derivatives to the basic static parameters.

2.1.3. Discrete Cosine Series Coefficients (DCSCs)

DCSC features [14] are similar to MFCC features. The features also imitate the perceptual characteristics of the human ear which resolves frequencies non-linearly across the audio spectrum. However, the features use a Discrete Cosine Series Expansion but applying it to feature trajectories over time. Fig. 1 illustrates the steps of computing DCSC features.

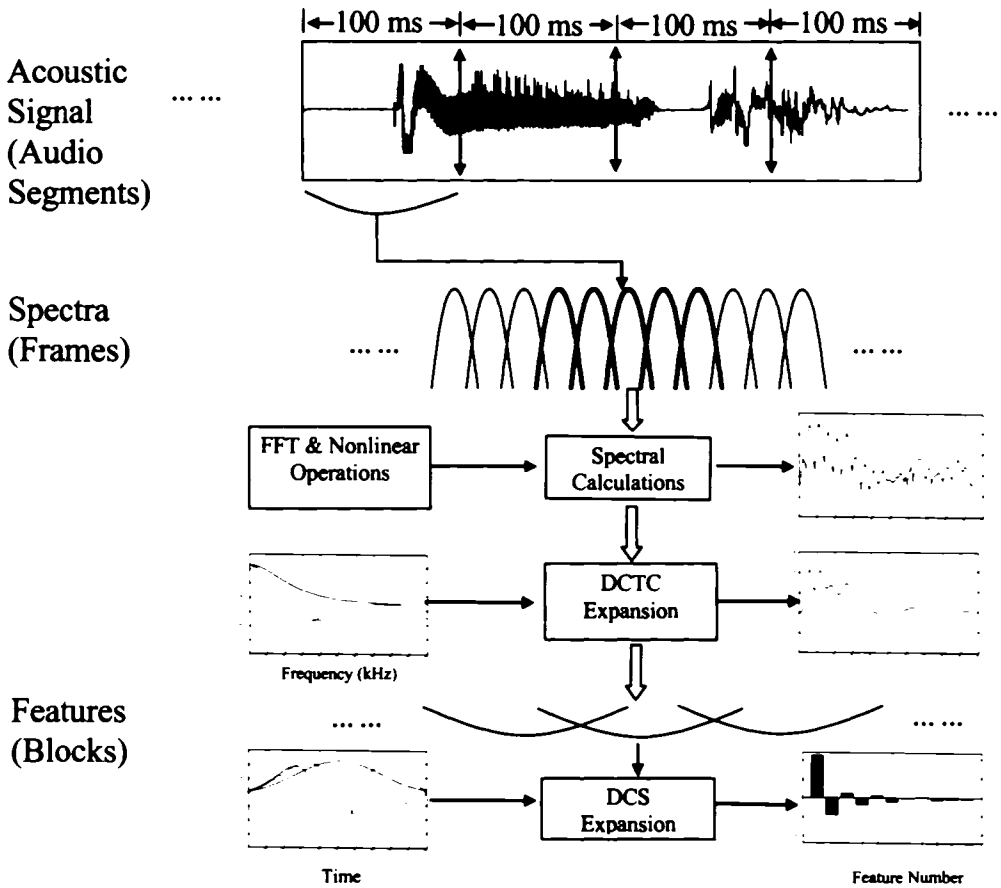

Fig. 1. Overview of signal processing for computing DCSC features.

The non-overlapping but continuous "segment" buffers form the interface between the data acquisition subsystem, such as a sound card, and the signal processing and recognition software. After being preprocessed with second order filtering, the processing follows the steps below:

1. The data in one segment is divided into overlapping frames as shown in Fig. 1. The number of frames is determined by the segment length, frame length and frame space.
2. Frames are multiplied by a Hamming window and an FFT is computed. Then the frequency data is morphologically dilated over frequency and logarithmically scaled for amplitude.
3. The features are based on a discrete cosine transform (DCT) of the log of the spectrum obtained in step 2, incorporating a frequency warping function $f' = g(f)$,

$$DCTC\ (i) = \int_0^1 X(f')\cos(\pi i f')df'. \tag{2.3}$$

By changing variables, the equation can be rewritten as

$$DCTC(i) = \int_0^1 X(f)\Phi_i(f)df \tag{2.4}$$

where the original cosine basis vectors are modified to account for the warping effect resulting in the basis vectors

$$\Phi_i(f) = \cos[\pi i g(f)]\frac{dg}{df}. \tag{2.5}$$

4. DCT is applied again over several frames, called a block. A block consists of an integer number of frames, with block spacing also being an integer number of

frame spaces. The number of frames in one block is called *block length.* In a similar manner, the DCTC terms for a block are also represented with a cosine expansion over time using:

$$DCSC(i,j) = \int_{-\frac{1}{2}}^{\frac{1}{2}} DCTC(i,t')\cos(\pi j t')dt'. \qquad (2.6)$$

The variable $t' = h(t)$ is considered to be a "warped" version of t where the function h is chosen to emphasize the center section of the segment. The time interval is normalized to [-1/2 ,1/2].

By changing variables, the equation can be rewritten as

$$DCSC(i,j) = \int_{-1/2}^{1/2} DCTC(i,t)\Theta_j(t)dt \qquad (2.7)$$

where the basis vectors $\Theta_j(t)$ are the modified cosines for the segment interval

$$\Theta_j(t) = \cos[\pi j h(t)]\frac{dh}{dt}. \qquad (2.8)$$

2.2 Review of speech training systems

2.2.1 Early attempts for speech training

Although considerable research effort has been devoted to speech training, very few speech training systems have become widely available commercially. Some systems [15] [16], using acoustic measures of speech performance, have the advantage of simplicity of operation and relative low costs, but are unable to give feedback for many aspects of production. While other physiologically-based systems [17] [18] overcome this limitation by supplementing the acoustic information with more sophisticated measurements such as airflow, laryngeal behavior and electropalatography. The

equipment used to obtain these measurements is often delicate, prohibitively expensive, and difficult to use without extensive training.

The success rate of most early systems [19] in speech articulation training for the deaf is unacceptably low. Using tests developed and conducted at NTID, less than 50% of freshman deaf students were found to have sufficient speaking skills and more than half of the passages they read aloud could be understood [20]. Detailed discussions of improved approaches to speech and written language instruction are given by [21] [22] [23]. The classic book by Ling [24] contains an excellent discussion of many complex and interrelated issues involving speech and the hearing impaired. In terms of language acquisition, Ling stresses the need to obtain skills at the phonetic level. Greater proficiency at the phonetic level increases the likelihood of organizing the basic sounds into a meaningful linguistic framework. In addition to the severely hearing-impaired population, there are many other children with speech or language disorders. For example, the National Center for Children and Youth with Disabilities that estimates there are over 1 million children in public schools with such disorders, with the largest number of these being phonological disorders. Speech language pathologists simply do not have the time to provide all the necessary one-on-one time with each child to overcome many of these problems.

2.2.2 Computer-based speech training aids.

Ever since the introduction of the first computer-based speech training aid about thirty years ago [25], the majority of speech training aids have relied on personal computers to implement complex speech processing algorithms and sophisticated yet appealing displays of speech. Examples of such aids include the Indiana Speech Training Aid

(ISTRA) [26] , Visual Speech Apparatus (VSA) [27], the commercially developed IBM Speech Viewer (Version III, 1997, newest version available [28]), Video Voice [29], Voice Prism (see Relevant Links in References [30]), Speech Training, Assessment, and Remediation (STAR) [31], Ortho-Logo-Paedia (OLP) project [32], and our own Vowel Articulation Training Aid (VATA) [33] [34]. Although many of the earlier versions of these aids required specialized high speed DSP cards (prior to about 1995), most of the current versions use only a "standard" PC with a microphone and sound card, thus making development and lower-system costs much less expensive and thus potentially resulting in widespread usage of the aids.

Some points from the literature more directly relevant to the work described in this dissertation are the following. Povel and Arends [35] share the view that, although the design and implementation of useful visual speech training aids is quite difficult, viable speech training aids will eventually be available and widely used. They mention that aids can be based on a *process* oriented approach *(*display of articulator parameters such as tongue or mouth position*)* or *product* oriented approach (display of acoustic information). They advocate the product-oriented approach, since it is far more convenient and appears to be the more natural way of learning speech. Anderson and Kewley-Port [36] evaluate three commercial automatic speech recognition systems for use in speech training with ISTRA. They conclude that an automatic recognizer's accuracy for normal speech does not correlate well with the recognizer's ability to evaluate speech quality. Of the three recognizers evaluated, only the DragonWriter, using HMMs, was very successful in classifying utterances as being either correctly or incorrectly produced. Moreover, Oster [37] states that visual aids must be acceptable and easy to use (hence, natural and

understandable) and pedagogically well designed, both from the viewpoint of the child and teacher; most importantly, the feedback should be given immediately without delay. Oster cites a case study whereby a 5-year-old, prelingually deaf child was successful in lowering his fundamental frequency from around 700 Hz to around 263 Hz after training with the Speech Viewer II. Mahshie [37] provides a review of Speech Viewer III. He concludes that the Speech Viewer does have several accurate and helpful displays for pitch, loudness, and timing training. Unfortunately the displays for phonemic training are neither very accurate nor easy to use, often even providing misleading feedback. Bunnell, et al, [31] summarizes a study which indicates that an HMM-based speech recognizer is able to provide reasonable good correlations to human ratings of /r/ quality from tokens produced by children with problems in productions of /r/ versus /w/. Cleuren [38] summarizes a few general principles with respect to the development of new training aid software, after interviewing people who regularly give therapy to hearing-impaired patients. These principles are: 1. Programs must be open. That is, users should be able to easily incorporate new customized material to the programs. 2. The program must be able to offer a variety of exercises with gradation in difficulty. 3. Feedback must consist of clear and sensible displays and comments that direct the learning process in the appropriate direction.

2.3 Vowel Articulation Training Aid (VATA)

In previous work at the ODU Speech Communications Laboratory, a visual speech training aid [33] [34] was developed for persons with hearing impairments using a Windows-based multimedia computer. The system provides real time visual feedback as to the quality of pronunciation for 10 steady-state American English monopthong vowels

(/aa/, /iy/, /uw/, /ae/, /er/, /ih/, /eh/, /ao/, /ah/, and /uh/). This training aid is thus referred to as a Vowel Articulation Training Aid (VATA).

The system has two main displays. One is a bargraph display, which gives feedback about how well speech utterances match discrete vowel categories. The other is an "ellipse" display, which provides more continuous feedback about vowel pronunciation.

The bargraph display (Fig. 2) resembles a histogram, with one bar for each vowel's sound of interest. The height of the vowel's bar varies in proportion to the accuracy of the speaker's pronunciation of that vowel. Correct pronunciation yields one steady, clearly defined bar, while the rest assume zero or small values. Incorrectly pronounced sounds may produce displays showing two or more partially activated category bars, no activated bars, or rapid fluctuations between bars.

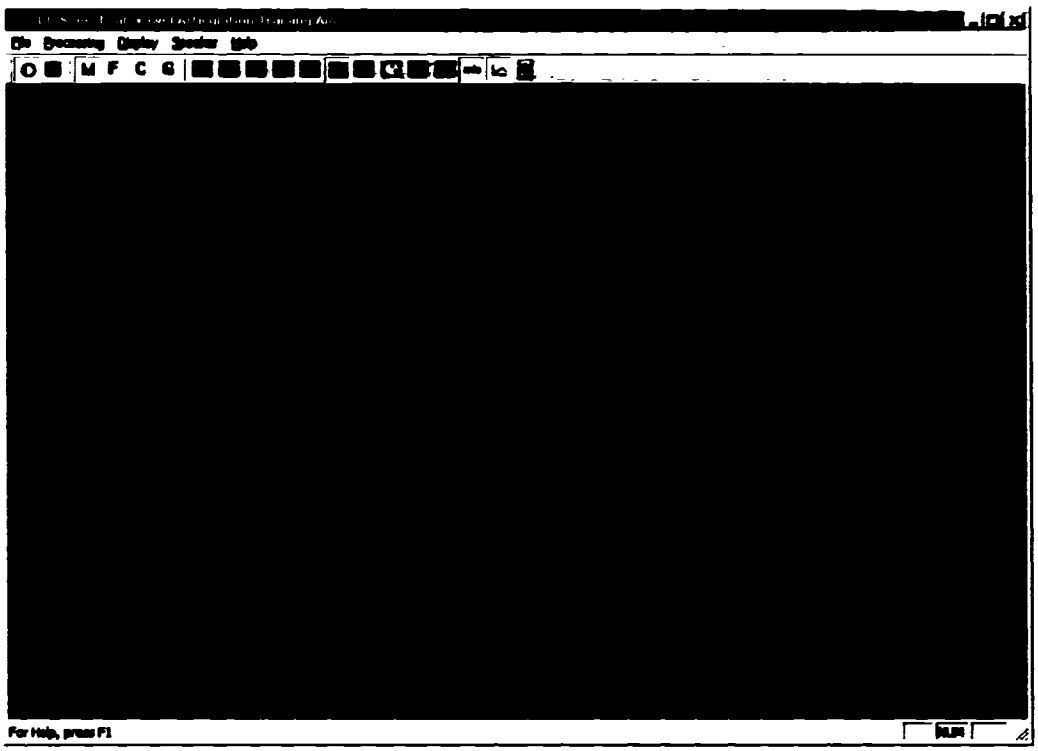

Fig. 2. Bargraph display showing response for correct pronunciation of /ur/.

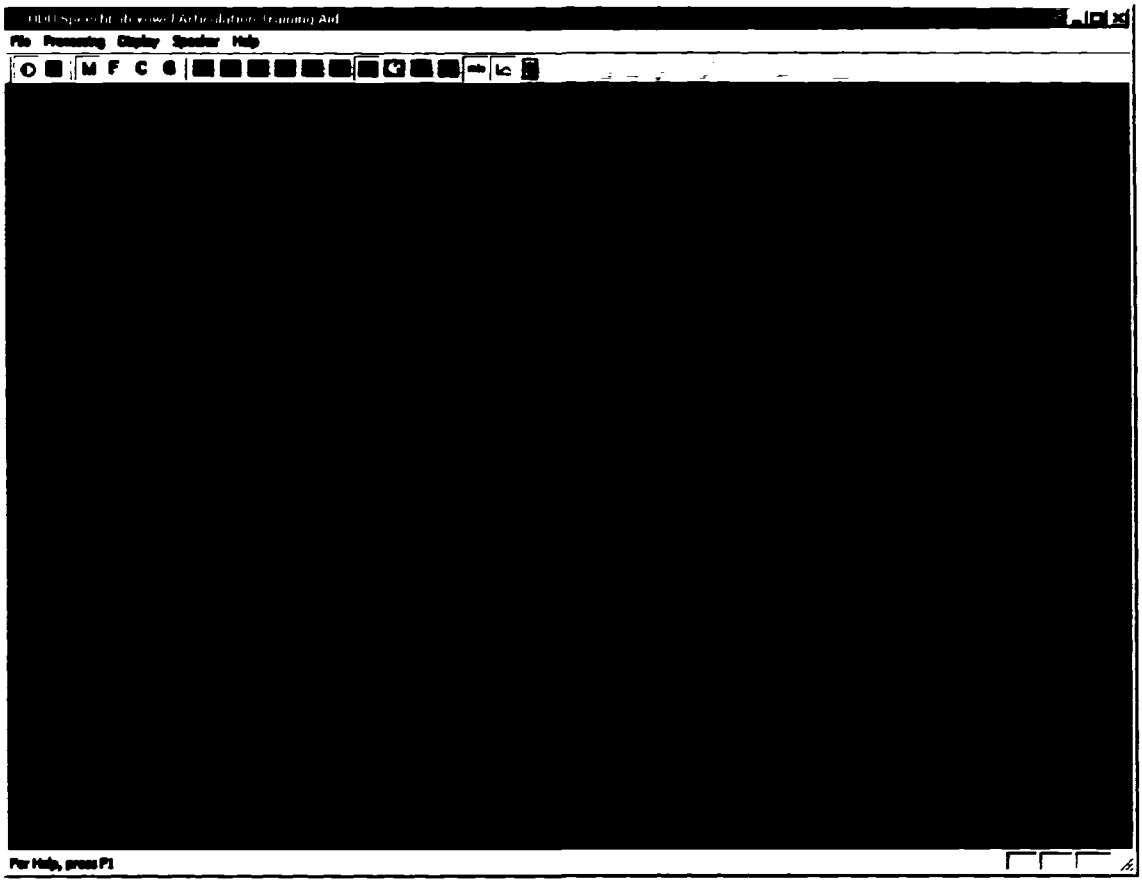

Fig. 3. Ellipse Display showing response for correct pronunciation of /uh/.

The ellipse display (Fig. 3) divides the screen into several elliptical regions, similar to an F1/F2 type display. Unlike the F1/F2 display, this 'ellipse' display bases its output on a neural network, which has been trained to transform DCTC features to specified target positions in a two-dimensional space. Correct pronunciation of a particular sound places a basketball icon within the corresponding ellipse and causes the icon's color to match that of the ellipse. Incorrect or unclear pronunciation results in the ball icon 'wandering' about the screen or coming to rest in an area not enclosed by an ellipse. By observing the continuous motion of the ball, a speaker hopefully can learn to adjust his or her pronunciation in order to produce the desired vowel sound.

In addition to the bargraph and ellipse display, three game displays have been developed. One game is a simplified version of "tetrus," for which one vowel sound is used to control the orientation of the colors in a falling bar in order to score. In another game, a chicken, controlled by two vowel sounds, attempts to navigate a busy highway without being hit by a vehicle. The third game is pacman, which uses four vowel sounds to control the four directions of movement of the game icon. Fig. 4 depicts the pacman game.

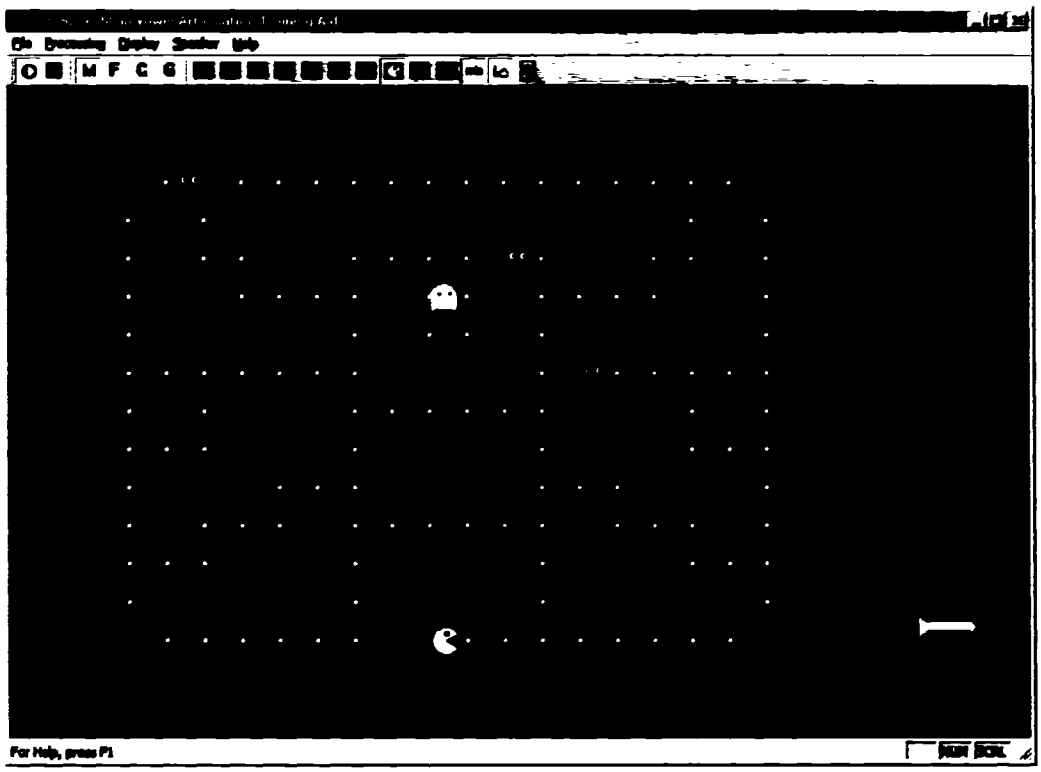

Fig. 4. Pacman game for vowel training.

A speaker group selection option with "CHILD," "FEMALE." and "MALE" settings allows all displays to be fine-tuned for better classification of sounds produced by child, adult female, or adult male speakers respectively. A fourth speaker group option, "GENERAL" uses a classifier based on all speakers.

Two neural networks are the primary classifiers used in VATA. One neural network, used as a neural network classifier for the bargraph display, has one input layer (typically 12 nodes, with one node per feature component), 15 to 25 nodes in the hidden layer and ten output nodes (one per vowel class). Another neural network has the same input layer as the first neural network, but with 2 hidden layers and 2 nodes in the output layer. The output values indicate the position in a two-dimensional space.

To overcome a major problem with the neural network approach for any type of pattern classification, for which "out-of-category" exemplars are also classified, according to whichever trained class is "closest" to the exemplar, a Gaussian Bayesian classifier was developed. In this classifier, a modified Euclidean-Distance measure has been incorporated with the neural network classifier as a secondary verification.

2.4 CVC Database preparation

VATA provides hearing-impaired people a tool to practice pronunciations of steady vowels. However, the goal is to develop a more general speech articulation training tool for people with hearing impairments. In this dissertation, the vowel training is extended to Consonant-Vowel-Consonant (CVC) short word training.

2.4.1 CVC Database Collection

A CVC token consists of a consonant, vowel and consonant pronounced together as a single short word. Actual speech productions also sometimes have a momentary silence (closure) in between the vowel and the final consonant. The consonants 'b','d','g','k','p','t,' classified as 'plosives' or stops, are produced by complete closure of the vocal tract followed by rapid release of the closure. Typically, this closure region appears as a momentary silence and should be labeled by the labeling algorithm.

However, the momentary silence is not always present, even for correctly sounding plosive consonants. In particular, for rapidly spoken utterances, such as some of those in the CVC database, the closure region is very short or even missing, and hence not feasible to label. The tokens recorded are listed in Table I as well as their pronunciations. For example "Bag" consists of 'b' (initial consonant), 'ae' (vowel) and 'g' (final consonant). Note that the presence of the closure (cl) region can also be labeled between the vowel and final consonant, but is not indicated as part of the pronunciation, since this closure is not phonetically significant. The labeling of the closure, when present, was however important, since information about the closure can be used to improve the performance of the automatic recognizer.

The database of CVC sounds in this dissertation was collected from adult males (120 speakers), adult females (142 speakers), and children between the ages of 6 and 13 (55 speakers). The tokens recorded are listed in Table I. This data was collected over a period of several years during which the sampling rate was increased from 11025Hz to 22050 Hz. The number of tokens is shown in Table II.

TABLE I

LIST OF CVC TOKENS RECORDED

CVC	Pronunciation	CVC	Pronunciation
Bag	b ae g	Boyd	b oy d
Bed	b eh d	Cake	k ey k
Beet	b iy t	Cot	k ao t
Bird	b er d	Cup	k ah p
Boat	b ow t	Dog	d ao g
Book	b uh k	Pig	p ih g
Boot	b uw t		

TABLE II

THE NUMBER OF SPEAKERS AND TOKENS IN THE DATABASE

Token Description	**Speakers / Recordings**			
	Male	**Female**	**Child**	**Total**
CVC Recordings	145/5566	166/6304	252/4665	563/16535

The usefulness of phonetically labeled acoustic speech data has long been recognized in the speech recognition research community. In fact, standards have been developed for creating and managing large labeled databases. The standards include the specifications for the acoustic files, including a certain type of header (NIST) and format for the actual samples (typically 2's complement binary numbers) and standards for the companion labeling files for each acoustic file. The acoustic file formats, with extension "wav", are similar but not the same as windows multimedia "wav" files. The phonetic labeling files are ASCII, and include the phone codes (using a two digit ARPABET alphabetic code to replace IPA notation) for each phone in the file, as well as the starting and stopping sample number for each phone. The CVC database was developed using these established conventions.

2.4.2 Database Segmentation

In addition to the database collection, each collected token was well articulated, properly endpointed 0, segmented with respect to phone boundaries. A semi-automated procedure was used to improve the quality and utility of the vowel tokens for use with the visual speech display. Those tokens which were not correctly classified by a neural network classifier were examined more closely by a human listener. Many of these tokens were either poorly pronounced or contained excessive noise. The real time vowel

display performed more accurately when trained only with the tokens remaining after the two-step reduction procedure.

A combination of automated and manual methods was used to segment the CVC database, which was followed by an examination for the correctness of pronunciation. For the process of phonetic segmentation, a preliminary segmental processing was performed using pitch and band energies for each token. The following multi-step approach was used for segmentation 0

1. A heuristic rule-based segmentation, based primarily on pitch, spectral band energies and endpoint detection was used as a first pass for segmentation.
2. An HMM triphone-based word recognizer was trained as a recognizer for the entire database, using the segmented data from step 1 to initialize HMM models.
3. Using the models trained in step 2, recognition was performed on all the training data. All CVCs that were not "correctly" recognized at this step were eliminated from the database.
4. HMM models were retrained using the data after step 3.
5. The HMM models obtained in step 4 were used for forced alignment of CVC tokens.

After those tokens considered "bad" were removed, a new set (subset of the original) of segmented CVCs is obtained as a new database. The number of tokens in this new database is listed in Table III

TABLE III

CVC DATABASE BEFORE AND AFTER REMOVAL OF "BAD" TOKENS

	Speakers / Recordings			
Token Description	**Male**	**Female**	**Child**	**Total**
CVC before bad data removal	145/5566	166/6304	252/4665	563/16535
CVC after bad data removal	145/4947	166/5793	252/4118	563/14858

2.5 Conclusion

This chapter presented the background of this dissertation. Firstly, three types of speech features - LPC, MFCC and DCSC were introduced. Then, some developed speech training systems at various locations in the world were reviewed. A vowel articulation training aid system (VATA), which was developed in the speech lab at Old Dominion University, was described in more detail since the CVC training system described in this dissertation is based on it. Finally, the CVC database used in this dissertation was discussed. The database segmentation was emphasized.

CHAPTER III

HIDDEN MARKOV MODELING FOR SPEECH RECOGNITION

3. 1 Introduction

Hidden Markov Models (HMMs) are the most popular models used to implement ASR systems. The primary reason for the widespread use is that an HMM has the wonderful ability to characterize the speech signal in a mathematically tractable way. HMM technology is the foundation of most successful speech recognition systems including many commercial products.

The principles underlying HMMs are given in 00. An HMM contains a hidden Markov chain and an observable process which is a probabilistic function of the states of the former. The fundamental assumption of an HMM is that the process to be modeled is governed by a finite number of states and that these states change once per time step in a random but statistically predictable way. To be more precise, the state at any given time depends only on the state at the previous time step. This is known as the Markovian assumption. Practically, the transition probabilities are organized in matrix form A, each element a_{ij} in A indicates the transition probability from state i to state j.

In most speech recognition systems, left-to-right HMMs are used. In a left-to-right HMM, only self-transitions and transitions to the next states are permitted. That it, the model has transitions that never go back to preceding states. The mathematical representation of this model is that in the transition matrix A

$$a_{ij} = 0, \qquad \text{for } i > j \text{ and } i < j-1 \tag{3.1}$$

An example of a 3 state discrete HMM is shown in Fig. 5 0:

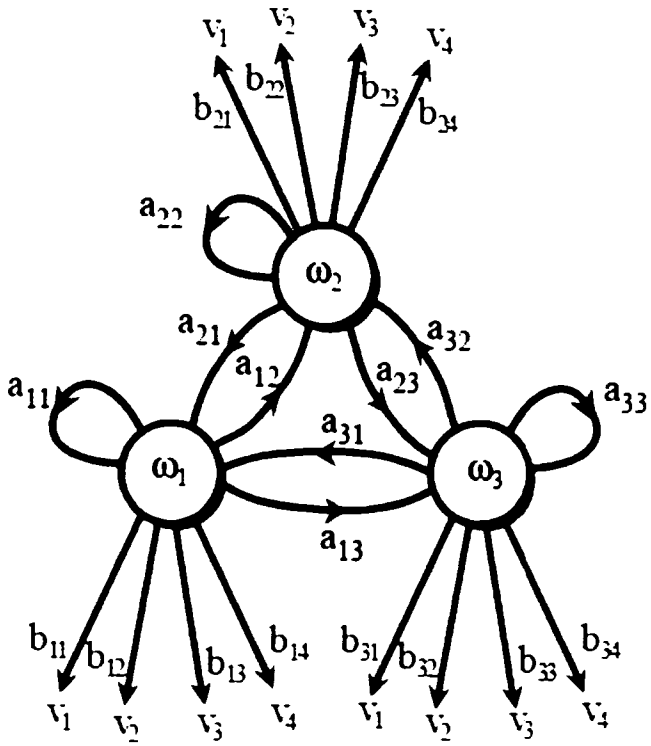

Fig. 5. An HMM with 3 hidden states and 4 visible states.

Ideally, there should be an HMM for every possible utterance. However, this is clearly infeasible for all but extremely constrained tasks; generally a hierarchical scheme must be adopted to reduce the number of possible models. In ASR systems, a HMM is usually used to model a speech unit. The speech unit could be a word, a phoneme or a syllable. When the vocabulary is small, it is possible to create an HMM for each word. However, if the vocabulary is large, it is unlikely that there is enough data to train all these word models, and meanwhile, the number of HMMs becomes too large to make training of HMM model parameters practical. Therefore, models for allophones of phonemes are considered at a different level of detail. The most commonly used units are speech sounds (phones) that are acoustic realizations of the linguistic categories called phonemes. Phonemes are speech sound categories that are sufficient to differentiate

between different words in a language. One or more HMM states are commonly used to model a segment corresponding to a phone. Then, word models consist of concatenations of phone or phoneme models (constrained by pronunciations from a lexicon), and sentence models consist of concatenations of word models (constrained by a grammar).

In a typical HMM-based ASR system, the HMM stage is preceded by the preprocessing (parameter extraction) stages. Thus the input to the HMM is a discrete time sequence of parameter vectors, such as those speech features described in the previous chapter. Each state j has an associated observation probability distribution $b_j(o_t)$ to an input feature vector at time t. An HMM can be classified as one of two types according to how the observations are characterized--one is the discrete HMM and another is the continuous HMM. In ASR, for the discrete HMM, vector quantization is used to associate each continuous feature vector with a discrete value. For the continuous HMMs, the observation distributions for the feature vectors are modeled by either a single or mixture of Gaussians. The probability $bj(o_t)$ of the symbol is given by

$$b_j(o_t) = \sum_{m=1}^{M_j} c_{jm} N(o_t; \mu_{jm}; \Sigma_{jm}) \tag{3.2}$$

where M_j is the number of mixture components in state j, c_{jm} is the weight of the m'th component and $N(\ )$ is a multivariate Gaussian with mean vector u and covariance matrix Σ.

Usually, correlation exists among speech features. In principle, it would be better to use full covariance matrices to represent the correlation of speech features. However, in many cases, especially when an HMM is complex (for example, the number of states and mixtures is large), computational, storage and robust estimation considerations make

the use of full covariance matrices in HMM observation distributions impractical. In practice, the features are generally assumed that they are uncorrelated and use a diagonal covariance matrix instead of a full covariance matrix.

The most popular of training algorithms for HMMs are the forward-backward (Baum-Welch) algorithm 0 and the Viterbi algorithm 0. Both algorithms are based on the general maximum-likelihood criterion.

3.2 HMM Algorithms

The theory and methodology for HMMs are described in many sources. The fundamental equation relevant for this process is a restatement of Bayes' rule as applied to speech recognition:

$$P(\mathbf{q} \mid \mathbf{O}, \lambda) = \frac{p(\mathbf{O} \mid \mathbf{q}, \lambda) P(\mathbf{q} \mid \lambda)}{p(\mathbf{O} \mid \lambda)} \tag{3.3}$$

where $P(\mathbf{q} \mid \mathbf{O}, \lambda)$ is the posterior probability of a hypothesized Markov model $\mathbf{q}$ given a vector sequence $\mathbf{O}$ and the parameter set λ. The posterior probability can not be computed directly, so this posterior probability usually is split into a likelihood $p(\mathbf{O} \mid \mathbf{q}, \lambda)$ that represents the contribution of the *acoustic model*, and a prior probability $P(\mathbf{q} \mid \lambda)$ that represents the contribution of the *language model.*

For speech recognition, the problems can be described as estimation of the posterior probability of HMMs given a set of acoustic feature vectors, where each HMM can be a phonetic, subword or word model. Once HMMs are trained (from a training data set), then an unknown acoustic feature sequence is used to estimate the most likely state sequences and the probability that the observations are generated by each of the

models. In the training step, the parameters of each model must be determined. HMM training and testing can be described of the following three problems:

3.2.1 The evaluation problem:

Suppose for a given HMM **q** with N states and parameter set λ, to calculate $p(\mathbf{O} \mid \mathbf{q}, \lambda)$ with a sequence of observations $\mathbf{O}(\mathbf{o}_1, \mathbf{o}_2, \cdots \mathbf{o}_T)$, a direct calculation involves a number of operations in the order of N^T. This calculation is unrealistic due to high computational complexity when T is large. Therefore, an efficient method is needed to reduce the complexity. A procedure, called the forward algorithm, is usually used to implement this task in a much simpler way.

In the forward procedure, a forward variable $\alpha_t(i)$ is defined as

$$\alpha_t(i) = P(\mathbf{o}_1 \mathbf{o}_2 \cdots \mathbf{o}_t, q_t = i \mid \lambda). \tag{3.4}$$

$\alpha_t(i)$ can be solved inductively, using:

$$\alpha_{t+1}(j) = b_j(\mathbf{o}_{t+1}) \sum_{i=1}^{N} \alpha_t(i) a_{ij}, 1 \le j \le N, 1 \le t \le T-1 \tag{3.5}$$

where,

$$\alpha_1(j) = \pi_j b_j(\mathbf{o}_1), 1 \le j \le N. \tag{3.6}$$

The recursive calculation is terminated with $a_T(i), 1 \le i \le N$, and then the required probability is obtained, which is,

$$P(\mathbf{O} \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i). \tag{3.7}$$

The complexity of this *forward algorithm* method is proportional to $N^2 T$, which is greatly reduced as compared to the direct computation.

In addition to the *forward algorithm*, a similar procedure called the backward algorithm is introduced. Similarly, a backward variable $\beta_t(i)$ is defined as:

$$\beta_t(i) = P(o_{t+1}o_{t+2}\cdots o_T \mid q_t = i, \lambda), \tag{3.8}$$

that is, the probability of the partial observation sequence from t+1 to the end, given states i at time t and the model λ. $\beta_t(i)$ can also be solved by a recursive method:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j), t = T-1, T-2, ..., 1, \quad 1 \le i \le N \tag{3.9}$$

with $\beta_T(i) = 1, \ 1 \le i \le N$, and

$$a_t(i)\beta_t(i) = p\{\mathbf{O}, q_t = i \mid \lambda\}, \ 1 \le i \le N, \ 1 \le t \le T. \tag{3.10}$$

This gives an alternate way to compute $p(O \mid \lambda)$ by combining forward and backfward variables, that is:

$$p(\mathbf{O} \mid \lambda) = \sum_{i=1}^{N} p(\mathbf{O}, q_t = i \mid \lambda) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i). \tag{3.11}$$

3.2.2 The decoding problem:

The decoding problem is to find the "optimal" state sequence associated with the given observation sequcen. The commonly used method to this problem is the Viterbi algorithm. That is, the whole state sequence with the maximum likelihood is found. In order to facilitate the computation, an auxiliary variable is defined as

$$\delta_t(i) = \max_{q_1, q_2, \cdots, q_{t-1}} P[q_1 q_2 ... q_{t-1}, q_t = i, \mathbf{o}_1 \mathbf{o}_2 ... \mathbf{o}_t \mid \lambda] \tag{3.12}$$

which gives the highest probability that partial observation sequence and state sequence up to $t= T$. When the current state is i, a recursive procedure is

$$\delta_{t+1}(j) = \left[\max_i \ \delta_t a_{ij}\right] \cdot b_j(\mathbf{o}_{t+1}) \tag{3.13}$$

where $\delta_1(j) = \pi_j b_j(\mathbf{o}_1)$, $1 \le i \le N$.

The procedure calculates $\delta_{t+1}(j)$ recursively with (3.12) until $\delta_T(j)$ is obtained. Then finally the state j^*, is found where

$$j^* = \arg\max_{1 \le j \le N} \delta_T(j). \tag{3.14}$$

Then starting from this state, the sequence of states is back-tracked as the pointer in each state indicates. This gives the required set of states.

3.2.3 The learning problem:

Suppose the structure of an HMM **q** and a set of acoustic feature vectors (training data) $\mathbf{O}(\mathbf{o}_1, \mathbf{o}_2, \cdots \mathbf{o}_T)$ are given, then a very important issue is how to adjust the HMM parameter set λ, so that training data is represented by the model in the best way for the intended application. There is no known method for obtaining the optimal or most likely set of parameters from the data, but a good solution by a straightforward technique can be nearly determined. The forward-backward method, which is also called Baum-Welch method, is usually used to estimate the parameters.

Firstly, this method uses two variables typically called $\xi(i,j)$ and $\gamma_t(i)$, where $\xi(i,j)$ indicates the probability of being in state i at time t, and state j at time t+1,

$$\begin{aligned} \xi_t(i,j) &= P(q_t = i, q_{t+1} = j \mid \mathbf{O}, \lambda) \\ &= \frac{\alpha_t(i) a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j)} \end{aligned} \tag{3.15}$$

and

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j). \tag{3.16}$$

With the above formulas, parameters π, A and B are estimated as

$$\bar{\pi} = \gamma_1(i) \tag{3.17}$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T} \gamma_t(i)} \tag{3.18}$$

$$\bar{b}_j(k) = \frac{\sum_{t=1,\ s.t.\ o_t=v_k}^{T-1} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}. \tag{3.19}$$

The learning procedure is iterative. In each step, $\bar{\lambda}$ is used to in place of λ and repeat the re-estimation calculation, until the parameters converge to some limiting points.

3.3 Hidden Markov Model Toolkit

The Hidden Markov Model Toolkit (HTK) 0 is a software package for building and manipulating hidden Markov models. Although it can be used for many purposes, that is, it can model any time series, in practice, it is primarily used for speech recognition research.

The toolkit provides several powerful tools that support speech signal processing, HMM construction and initialization, HMM training, testing and result analysis. HTK also supports multi-purposes for speech recognition applications. For example, users can use the HTK to build isolated words recognizers; users can also build recognizers for large vocabulary continuous speech recognition.

The HTK consists of a set of library modules and tools available in C source form. The tools provide sophisticated facilities for speech analysis, HMM training, testing and

results analysis. The software supports HMMs using both continuous density mixture Gaussians and discrete distributions and can be used to build complex HMM systems. In addition, the HTK provides many standard file formats, and consists of a set of library modules and tools available in C source form, which make it possible to interface to other software packages.

The uses of HTK in this dissertation include:

1. Creating isolated words recognizers with the CVC database. HMMs were trained with both of MFCC features and DCTC features. Then the recognition rates with recognizers were used as baselines to compare with other algorithms.
2. Each token in CVC database was segmented with the HTK. Combined neural network and HMM methods were investigated to improve recognition performance. Since the neural network training requires specifying the target of each speech feature vector, tokens in the database must be segmented to determine the targets.
3. The HTK front-end analysis tool was substituted with our program, which normalized speech features with a neural network. Then another HTK configuration was used to create the recognizer.

3.4 Experiments and Evaluation

The experiments reported in this section were performed with the HTK toolkit. The HMM recognizers were trained and tested with MFCC features and DCSC features respectively. The recognition results were used as the baseline to compare with other algorithms which are described in later chapters.

3.4.1 Database division

The entire CVC database was divided into 2 sets: 7000 tokens were used as the training set and the remaining 7858 tokens were used as the test set. The division is based on the considerations listed below:

1. The goal of this dissertation is to build a speech training system which can adapt to all speakers, so the speech engine is required to be speaker independent. The speakers in the training set do not appear in the test set.
2. To test the generalization of the recognizers, the test set cannot be too small. In this dissertation, more than half the tokens were used as the test set.
3. This division is only to evaluate the performance of the algorithms; the actual recognizer was trained with all tokens since the recognizer will be more robust with more training data.

TABLE IV

DIVISION OF THE CVC DATABASE

Training Set				**Test Set**			
Male	Female	Children	Total	Male	Female	Children	Total
1435	3561	2004	7000	3826	1918	2114	7858

3.4.2 Speech features

In this section, recognition results were obtained using MFCC and DCSC features. An HTK front-end tool – Hcopy was used to compute 39 MFCC features (12 MFCC parameters plus energy and their first and second derivatives). The features were

extracted from each 25-ms frame. The frame spacing was 10 ms, and 26 filter channels were used.

Several experiments were conducted with DCSC coefficients, varying the block length and number of DCS terms (i.e., the time interval for a block and the degree of time resolution). The methods of computing DCSCs were described in chapter 2. The parameters other than block length and number of DCS terms were fixed as: the signal was pre-filtered with center frequency 3200 Hz. The frequency spectrum was processed for the frequency range 100 Hz. to 5400 Hz. The frame size was 25 ms, the frame spacing was 10 ms and the DCTC warping fact was set to 0.4. The number of DCTC terms was 12, but the actual dimension of the speech vector was the number of DCTC terms times the number of DCS expansion terms.

3.4.3 Dictionary

The dictionary consists of words and their pronunciation. There are totally 13 CVC words in the database. The basic pattern of pronunciation for each word is Consonant-Vowel-Consonant. Considering there could be a closure between the vowels and final consonants, each word is assumed to have two possible pronunciations as follows: Consonant-Vowel-Consonant and Consonant-Vowel-Closure-Consonant. Meanwhile, a symbol named "Sil" indicating silence that occurs at the beginning and end of the CVC words was also included in the dictionary.

3.4.4 Grammar

For each speech recognition system, a grammar is usually applied to restrict the search to allowable phrases. It can improve the recognition rate and speed, especially for large vocabulary continues speech recognition tasks. The grammar for these experiments

is very simple since the recognizers are only for isolated words recognition and the vocabulary is small (Only 13 words). Each input speech acoustic sequences are assumed to represent a CVC word with preceding and following silence. Therefore, the grammar for each token is:

silence + word + silence.

3.4.5 Phone models

Two types of phone models were created in the experiments, monophone models and triphone models. As described in chapter 2, there are 21 monophones (6 consonants, 13 vowels, 1 silence and 1 closure). Each monophone was modeled by a left-to-right hidden Markov model with 3 states. There are 5 Gaussian mixtures per state with a diagonal covariance matrix. An example of a model is shown as Fig. 6.

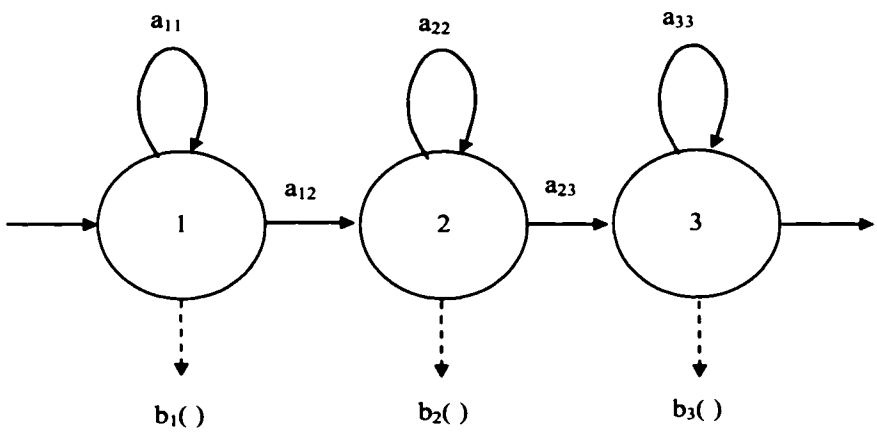

Fig. 6. A left-to-right HMM.

A speech recognition system can obtain better performance with triphone models than monophone models since the triphone models contain more contextual information which is very helpful for speech recognition. Triphone models can be made by tying a

monophone model with its adjacent models. In this case, 71 triphones are needed after considering all the relevant combinations of phones.

3.4.6 Training and Testing

The training process began with monophone models, using HTK tools for this process. First, each model was initialized using the Viterbi algorithms with the training database. The time labeling of each phone was used. The parameters of each model were estimated from the data that belong to its class. Then each initialized model was re-estimated with the Baum-Welch algorithm. Next, all models were trained with the Baum-Welch algorithm in the embedded mode for 3 iterations.

The next step was to create context dependent triphone models by copying the parameters of trained monophone models. Once again, the parameters of each triphone model were trained with with the Baum-Welch algorithm in the embedded mode for 3 iterations.

The Viterbi algorithm was used in the recognition process. The test set, which includes 7858 tokens, was processed by a HTK general purpose Viterbi word recognizer – Hvite. It matches a speech file against a network of HMMs and outputs a transcription for each. When performing N-best recognition a word level lattice containing multiple hypotheses can also be produce.

The final results were reported in terms of percent Word Error Rate (WER) on the test data. WER is defined as follow:

$$WER = \frac{N - H}{N} \times 100\% \tag{3.20}$$

where H is the number of tokens that correctly recognized and N is total number of tokens in the test database.

3.4.7 Results

As described above, two sets of experiments were performed with respect to speech features. The results are shown as Table V, as percent of WER.

TABLE V

HMM WER FOR CVC DATABASE WITH MFCC AND DCSC FEATURES

		Monophone Models		**Triphone Models**	
		Training Set (%)	**Test Set (%)**	**Training Set (%)**	**Test set (%)**
MFCC Features		4.1	6.0	1.8	5.1
DCSC Features with DSC=3	bk_len = 5	5.5	9.5	2.2	6.9
	bk_len = 10	4.5	8.1	1.8	6.0
	bk_len = 15	4.2	8.3	1.8	5.9
	Bk_len = 20	4.6	9.2	2.2	7.3
	Bk_len = 25	5.0	10.2	2.6	8.5
	Bk_len = 30	6.2	11.6	3.4	9.6
DCSC Features with DSC=4	bk_len = 5	5.8	10.0	2.4	7.1
	bk_len = 10	4.4	8.3	1.8	5.8
	bk_len = 15	3.8	7.6	1.7	5.3
	Bk_len = 20	3.8	7.1	1.6	6.6
	Bk_len = 25	4.1	7.3	1.8	6.4
	Bk_len = 30	5.3	9.3	2.5	7.7

In Table V, bl_len refers the block length indicating the number of frames per block. All numbers in the table indicate percentages. From this table, the best test performance was 5.1% WER achieved by triphone models with MFCC features. In general, no matter which training set or test set was used, the accuracy obtained with triphone models is always higher than that obtained with monophone models. As for DCS features, the best performance was a WER of 5.3% achieved with a block length of 15

and 4 DCSC terms. The conclusion can be drawn that performance varies with block length for this method. However, larger block lengths do not always result in better performance; furthermore longer block lengths result in more latency in response times of the system. In the actual CVC system, several factors must be considered to choose a suitable block length.

3.5 Chapter Conclusion

This chapter introduced the concept of HMMs and algorithms related to HMMs. A brief description of the HTK also was given. The HTK is a powerful toolkit for building and manipulating HMMs. Several experiments were then performed with the HTK and the results were given. Although these results are compared, the real goal is not to compare results for the settings tested in this chapter, but rather to use these results as a baseline for comparison with methods presented in later chapters.

CHAPTER IV

COMBINED NEURAL NETWORKS AND HIDDEN MARKOV MODELS FOR AUTOMATIC SPEECH RECOGNTION

4.1 Disadvantages of HMMs

Although the HMM approach is the mainstream one used for state-of-the-art speech recognition systems, there are still several problems with HMMs. These problems include:

1. Some assumptions are not reasonable

   To reduce the complexity of ASR systems, some assumptions are made when applying HMMs. However, not all assumptions are valid for speech. For example, the models assume that the emission and the transition probabilities depend only on the current state. However, the next state may depend on the past several states. Gaussian Mixture Models (GMMs) are generally used for continuous-density HMMs for speech recognition systems. When using GMMs, usually a predefined number of mixtures is chosen and each mixture component is assumed to be distributed as a normal random variable. These assumptions are primarily based on practicality (i.e., managing computational and overall complexity) and are not theoretically well founded for speech. For examples, feature components are assumed to be independent of one another, primarily so that covariance matrices are diagonal, greatly reducing the number of parameters that must be estimated, rather than confidence that the features are actually independent. Note however, that better ASR performance is generally obtained using diagonal covariance matrices rather than full covariance matrices, since generally not

enough speech data is available to reliably estimate the feature covariances for GMMs.

2. HMMs only use within-class data for training. That is, the data used to train a HMM all belong to this class, while all other data are ignored by the model. HMM training is based on the Maximum Likelihood training criterion. These lead HMMs to have poor discrimination between the acoustic models.

4.2 Neural networks for classification

Neural network classifiers are a powerful technique and they have been successfully applied to many fields, including automatic speech recognition 0 00. The elemental units in the neural networks are called neurons, which mimic the behavior of human biological neurons. Each neuron has many inputs and one output. The output is usually a nonlinear mapping of a summation of the weighted inputs and a bias. A neural network consists of many layers--an input layer, one or more hidden layers and an output layer. The neurons are interconnected with other neurons in the same or different layers. A neuron's connection topology with other neurons may vary from fully connected to sparsely or even locally connected.

The advantages of the neural networks can be summarized as:

1. Adaptive learning: A neural network has strong adaptive learning ability, that is, it can learn to perform tasks based on the data given for training or from initial experience.
2. Real Time Operation: NNs are parallel structures, which make it is easy to design and manufacture special hardware to achieve real time computation.

3. Flexibility: NNs can handle input data with samll changes in the input data such as added noise.
4. Fault Tolerance: NNs store redundant information, so partial destruction of the neural network does not completely damage the network response, but may degrade performace.

The process of neural network training is to adjust weights to minimize error between output nodes and the neural targets for inputs. Until now, the most successful neural network training algorithm is the backpropagation (BP) algorithm 0, which essentially is a gradient search method, especially arranged to match the architecture of a neural network. The training process works in iterative steps: First, the network is initialized with random weights. Then, for a given input, the differences between output values and desired values (targets) are calculated as errors. The error values are then propagated backwards through the network, and small changes are made to the weights in each layer. The weight changes are calculated to reduce the error signal for the input that has been applied. The whole process is repeated for each of the points in a database. The cycle is repeated until the overall error value drops below some pre-determined threshold. At this point the network is said to have learned the problem "well enough." Depending on the complexity of the problem, the network may not learn the ideal function, but generally it will at least asymptotically approach the ideal function.

Because neural networks are good at pattern recognition and ASR is basically a pattern recognition problem, it is natural to consider applying neural networks to speech recognition. The first attempt to use NNs for ASR was begun in the 1980s 0, and

followed by [51]. There are obvious advantages for applying neural networks to speech recognition. These include:

1. Discrimination: Neural networks are good at discrimination. Discrimination will be maximized between classes, rather than most closely matching the distributions within each class.
2. Neural networks can handle complex probability densities. Since neural networks do not require any knowledge of the distributions of training data, nor make any assumptions about these distributions, the models do not need to be predefined. On the other hand, for N-category classification problems, one neural network with N output nodes is able to recognize N patterns, whereas HMMs must create at least N models.

Researchers [52] have built phoneme and words recognizers with NN technology. There are two basic approaches to speech classification using neural networks: static and dynamic. Neural networks used for the static classification approach accept all of the inputs at once and generate the corresponding output. This approach requires that all utterances have the same length. By contrast, in dynamic classification, the inputs to neural network is a small window of the speech, and this window slides over the input speech while the network makes a series of local decisions, which have to be integrated into a global decision at a later time.

The neural network architectures used for speech recognition are usually one of the following three types, as summarized below:

1. Feed-forward neural networks [53]: The architecture of these neural networks is the one most commonly used for pattern recognition and it is the basis for other

neural networks. It is also called a Multilayer Perceptron (MLP). A Feed-forward neural network has a layered feed-forward architecture with an input layer, zero or more hidden layers, and an output layer. The nodes in each layer only connect with their adjacent back layer and forward layer. There is no feedback in the network. Each layer computes a set of linear discriminate functions followed by a nonlinear function.

2. Time-delay neural networks (TDNNs) [54] [55]: The networks are designed to accommodate a time sequence of features. The architecture is quite similar to a feed-forward neural network, consisting of input, hidden and output layers. However, each hidden unit accepts inputs not only from the current time-slot features, but also can accept previous time-slot features. Hidden units at "delayed" locations accept inputs from the input layer that are similarly shifted. The difference of the training process for BP algorithms is that a constraint is added that corresponding weights are forced to have the same value. An example of a TDNN is shown as Fig. 7

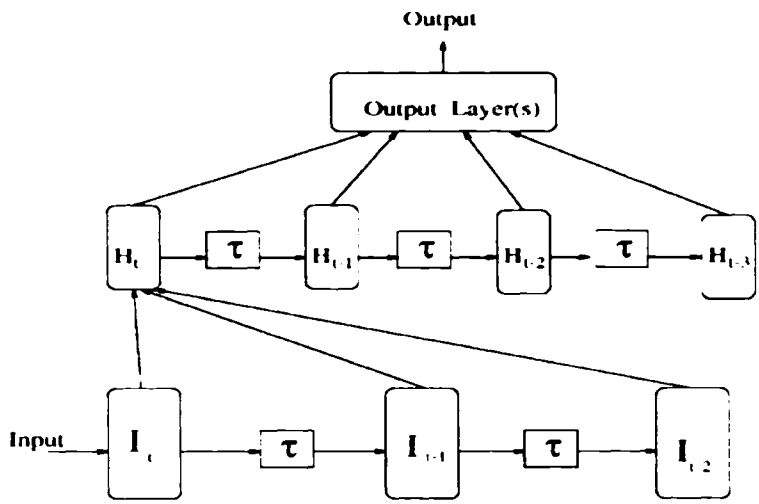

Fig. 7. A time delay neural network.

3. Recurrent neural networks (RNNs): [56] A recurrent neural net is a neural net with a ``feedback" loop in it. There are several architectures for recurrent neural network. Fig. 8 shows an example of an RNN [57].

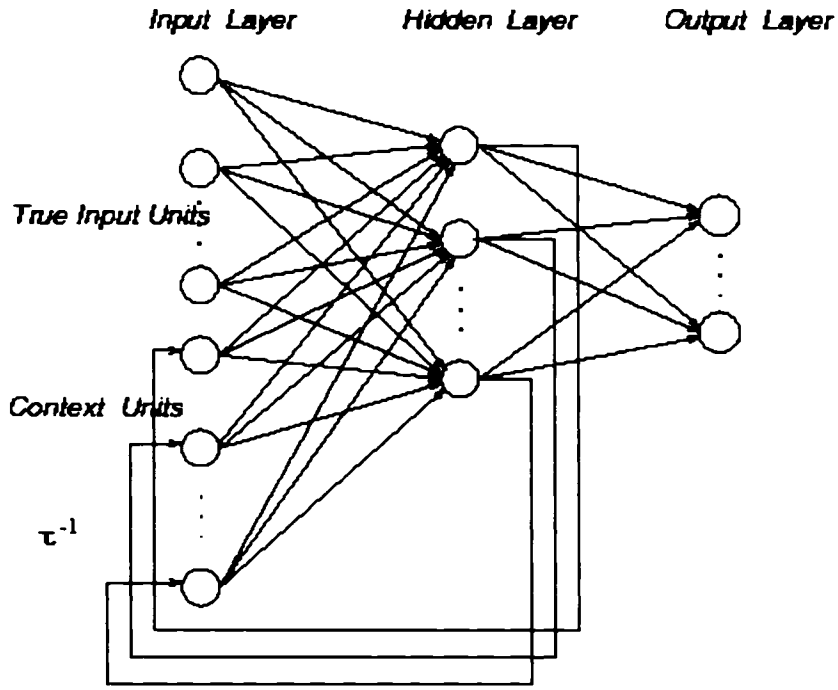

Fig. 8. A simple recurrent neural network.

In Fig. 8, at each time step, a copy of the hidden nodes is taken and stored in the context units. Then in the next time step, the input vector along with context units are input into hidden nodes. The context can be connected to the hidden units in many ways. The network is trained using BP through time. This method has proved to work as well as the MLP approach, but with even fewer parameters. The principal drawback appears to be a somewhat tricky training schedule, due to potential instabilities.

Although some speech recognizers based on neural networks have been developed, the systems are usually for phoneme or word recognition. There are no

successful systems using only NN technology for continuous speech recognition. The major impediment is that NNs are not good at processing complete temporal sequences such as speech signals. In contrast, one of the major advantages of HMMs is their convenient structure for processing temporal sequences.

4.3 Introduction of hybrid models

The idea of hybrid systems comes from the idea that HMMs and neural networks can overcome each other's limitations and take advantage of the strength of each. Some researchers [3][58][59] have investigated hybrid methods in detail. In this section, four major categories of hybrid methods are summarized. The method of using neural networks as statistical estimators is emphasized because it is the method used in this dissertation.

4.3.1 Using ANNs to simulate HMMs

This approach was popular around the early 1990s. The primary idea is to use NN architecture to mimic an HMM. This method was introduced by Lippmann and Gold in 1987 [50]. They proposed a recurrent neural network to mimic a Viterbi network, which was implemented in VLSI, and used to recognize isolated words. However, the network does not have a training procedure and can only decode acoustic features. The training procedure is accomplished using the Baum-Welch algorithm to initialize the network's parameters.

In 1990, Bridle developed a connectionist architecture [60], called alpha net, which was able to behave like an HMM. Like the Viterbi net, it is also a recurrent network. However, it integrates the Baum-Welch algorithm into the network, so it can be used for training.

These approaches gave the idea that the neural networks can be combined with HMMs for ASR. However, since they simulate the HMM, the limitations for HMMs still exist in these approaches.

4.3.2 The "global optimization" approach

This approach was first proposed by Bengio in the 1990s 0. The idea was inspired by the Alpha net. Here, the ANN plays the role of a feature extractor for a standard continuous density hidden Markov model (CDHMM).

In the CDHMM, the acoustic features are assumed uncorrelated, which is known to be untrue. However, ANNs can reduce the correlations among the features and can be used to transform the input acoustic features to compact, low-dimensional, discriminating features. These new features can be used with a standard CDHMM and the performance potentially improved.

Experimental results showed that training the ANN jointly with the HMM in the proposed hybrid architecture improved recognition performance over the standard CDHMM.

Riis and Krogh proposed another hybrid model, which is called Hidden Neural Networks (HNN) [61]. In this model, they extend the application of ANN from feature extraction to the estimation of HMM parameters. All parameters in the HNN are estimated simultaneously according to the discriminative conditional maximum likelihood criterion. It achieves better performance than the standard HMM on the task of recognizing broad phoneme classes for the TIMIT database.

4.3.3 Networks as vector quantizers for discrete HMMs

In this approach, ANNs are used as an alternative method to standard clustering algorithms to generate codebooks for discrete HMMs. Here, a neural network is an optimal neural vector quantizer.

[62] and [63] proposed this hybrid system and explained the principles and motivations. In the system, a feed-forward net was trained to perform vector quantization for a discrete HMM. The training procedure is an unsupervised algorithm whose goal is to maximize the mutual information between the input classes of the network and its resulting sequence of firing output neurons.

The primary advantage of this system is that such neural networks can be easily combined with HMM's of any complexity with context-dependent capabilities. Some experimental results were reported for the ATR speech database. It showed a 25.0% relative error reduction over VQ based on k-means cluster.

Jang and Un [64] proposed another similar hybrid system for SI isolated words recognition. The system constructed a fuzzy-vector quantizer (FVQ) using TDNNs, and then fed the outputs of the FVQ to a discrete HMM. In this method, the TDNNs and HMM were trained separately using standard algorithms. The experimental results for an SI database, which is an isolated Korean words database, showed 44.9 % relative WER reduction with respect to a standard discrete HMM.

4.3.4 NNs as Statistical Estimators

This approach was first proposed by Bourlard et al in 1993 [65]. The basic idea is to regard the neural network outputs as posterior probabilities, and then to use Bayes' theorem to transform them to observation probabilities. Finally, the transformed

probabilities are used as HMM emission probabilities, rather than using the Gaussian Probability Density Function (PDF) estimations.

A typical architecture of this hybrid system can be represented as Fig. 9 [66]

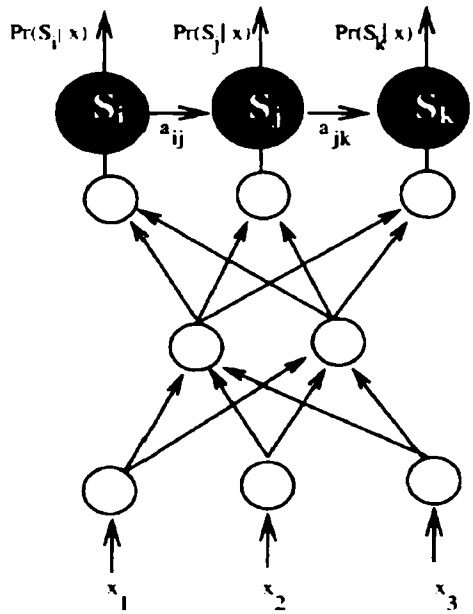

Fig. 9. Basic hybrid architecture on NN as a statistical estimator.

In Fig.9, the HMM is a left-to-right model with hidden states S_i, S_j and S_k. An acoustic feature vector $x = \{x_1,x_2,x_3\}$ is the input to a trained 2-layer feedforward neural network and the outputs are used as HMM posterior probabilities.

A brief proof on why the outputs can be interpreted as estimates of a posteriori probabilities is given in [67]. According to the proof, this approach has two requirements for the ANN training.

1. The system must be sufficiently complex to be able to form a good approximation of the mapping function between input and the output class.
2. The system must be trained to a global error minimum.

The emission probabilities can be computed using the equation below 0:

$$p(\mathbf{x}|S_i) = \frac{p(S_i|\mathbf{x})p(\mathbf{x})}{p(S_i)}. \quad (4.1)$$

In (4.1), S_i is the ith hidden state in the HMM which can represent a phoneme or a sub-phoneme. p(x) is the observation probability which is equal for all states and can therefore be skipped. $p(q_i)$ can be computed from relative frequencies in the training data.

A central issue with this approach is the training procedure. As is well-known, the training procedure for neural networks in this approach is supervised training, which requires that both the inputs and the outputs are provided. Morgan and Bourlard suggested an iterative training procedure [3], which start with an initial segmentation of the acoustic observations. The initialization may be obtained by utilizing a standard HMM process, also referred to forced alignment, or by dividing the observation sequence into equally sized segments.

The training processing consists of first training an NN with a labeled training set, and using the resulting recognition rate as a baseline score. The NN outputs are then used as inputs to a left-to-right HMM after converting them to the emission probabilities using (4.1). Then the Viterbi algorithm is applied to determine the new labels for the training set. Finally, an iterative procedure is executed until the recognition score is not improved.

Morgan and Bourlard successfully applied this hybrid system to the SRI's DECIPHER system and achieved a 5.8% word error, as opposed to an 11% error rate for a context-independent version with pure HMMs.

Some other researchers have used different neural network architectures with HMMs for speech recognition. For example, Singer and Lippmann [68] used Radial

Basis Function (RBF) networks instead of MLPs as posterior probability estimators. Robinson et al. [69] used recurrent neural networks as estimators. In [54], TDNNs were used as posterior probability estimators.

Some other significant attempts based on this method were also done by other researchers. In [70], Yan et. al. tried to use a continuous number as the targets of neural networks instead of using binary values "0" and "1". Kirchhoff and Bilmes [71] proposed a method that combines different architects' neural network outputs as the posterior probabilities..

4.4 The hybrid method used in this dissertation

4.4.1 Hybrid method selection

In this dissertation, hybrid methods are used to build a speech recognizer based on the following considerations:

1. Better performance: The goal of this dissertation is to develop a system which is helpful for people with hearing impairments to practice their pronunciation. Speech recognition technology will be used as the centerpiece for the system. The speech recognizers will tell users if their pronunciations are correct or not. Since the recognizers act as a "teacher", they should have knowledge of the teaching materials. That is, the recognizers must have high recognition accuracy; then, the users can get benefit from the system. In theory, the hybrid method of using NNs as statistical estimators has better performance than HMMs only, so this method was selected in this dissertation.
2. Phoneme level recognition: The system is designed for articulation speech

training. The recognizers should recognize user's speech at both the word level and phonetic level. When a user practices pronunciation for a given CVC word, the system is required not only to tell the user if his/her pronunciation for this word is correct or not, but also is able to tell the user if his/her pronunciation is correct for each phoneme in the word. Note that the goal of designing speech training recognizers is different from other common speech recognizers, such as command recognizers or dictation recognizers. Typical recognizers only indicate the closed-set matching of input speech signals to models. The recognizers are required to adapt to the user's behavior. Even if some pronunciations are only partially correct, the system will, ideally, eventually learn to accept them. In contrast, for speech training tasks, the users are required to adapt to standard pronunciations. The computer-based training systems should be able to visually indicate the users' pronunciation error at both the word level and the phonetic level. Ideally the errors should be presented in a manner that guides the user to a correct pronunciation. HMMs with Gaussian mixture models can provide phoneme recognition results, as well as word recognition results. However, neural networks have more advantages for phoneme level recognition since the results are more discriminative.

3. Real-time requirements: The articulation training process for the human user appears to be more natural and easy to interpret if the computer-based training system presents visual indicators of the speech with minimum time delay. Compared to pure HMM methods, neural networks can give more rapid responses due to its parallel computation ability at the phonetic level.

After comparing several hybrid of HMM/NN methods, the method of using the neural networks as statistical estimators is selected in this dissertation. This method, according to many studies in the literature [3] and experiments in this dissertation, has better performance than the HMM method only for word level accuracy. Additionally, this method is easier to implement for the tasks investigated in this dissertation. The neural network outputs, which are used as the observation probabilities of HMMs for word recognition, are also used as phonetic recognition results for the system. Other hybrid methods have some limitations: for example, the method of using neural networks to simulate HMMs only simulates HMMs with the networks and does not appear to have an advantage over HMMs alone; the "global optimization" approach converts speech features from one feature space to another space, but it does not determine the phoneme recognition; the method of using Networks as vector quantizes for discrete HMMs has the same limitation as the "global optimization" approach.

4.4.2 Neural Network Architectures

Three neural network architectures are mainly used in speech recognition tasks; they are feed-forward neural networks, time delay neural networks and recurrent neural networks. In this dissertation, the experiments were based on feed-forward neural networks and time delay neural networks. As for recurrent neural networks, the architectures are more complex and even the training of these networks may not stabilize. In addition, some researchers [58] found no significant difference in the recognition accuracy of recurrent neural networks and other neural networks. They also conclude that

it is important only that a network have some form of memory, regardless of whether it's represented as a feed-forward input buffer or a recurrent state layer.

The neural networks in this dissertation have three layers: an input layer, a hidden layer and an output layer. The first layer is an input layer and the number of input nodes is determined by the network's architecture. For feed-forward neural networks, the number of inputs nodes is the dimension of speech feature vectors; for the time delay neural networks, the number of inputs nodes is determined by both the dimension of speech feature vectors and the number of time delays. The actual value is product of the dimension of speech feature vectors and number of time delays. In theory, the time delay neural network works better than the feed forward neural network because the inputs contain more context information. That is, the performance should be better with more time delays. However, in practice, this context information appears in the form of a higher dimensionality input to the neural network and may result in problems referred to as the "curse of dimensionality." That is, much more training data is needed for this extra dimensionality and the neural network training may not be representative of text data. With a fixed amount of training data, beyond a certain limit, this extra context information actually reduces recognition discrimination.

In optimizing the design of a neural network, an important consideration is whether the neural network should have a hidden layer, how many hidden layers are suitable for this particular case and how to determine an appropriate number of hidden nodes. A neural network with no hidden layers can form only simple, connected decision regions. This dissertation did not consider this neural network architecture since it is known that the speech pattern recognition is highly nonlinear and complex. Architecture

with more than one hidden layer greatly increase training time and complexity; also it has been shown that any function that can be computed by an MLP with multiple hidden layers can be computed by an MLP with just a single hidden layer if it has enough hidden nodes. How to determine the number of hidden nodes is also an important consideration. A small number of hidden nodes reduces the network's computational complexity. However, the recognition accuracy is often degraded. The more hidden nodes a network has, the more complex decision surface can be formed, and hence the better classification accuracy it can attain. However, too many hidden nodes will increase the training time and potentially reduce the network's generalization, which means the performance may be considerably lower for test data than for the training data. Generally, the number of hidden nodes is empirically determined by a combination of accuracy and computational considerations, as applied to a particular application. Later in this chapter, experimental results are given that show recognition accuracy as a function of the number of hidden nodes.

4.4.3 HMM architectures

The HMM architectures in this work are simple left- to-right models. Each model, which indicates a phoneme, has only one state. Although some researchers [72] showed that a model with more than one state outperformed the one state model, Only one state was used for each model in this dissertation because:

1. Although more states per model may improve word recognition performance, it will degrade phoneme recognition accuracy since there are more categories for neural network training.

2. The CVC training system requires that the speech engine gives real time response at the phoneme level for each input frame. If the phoneme model has more than one state, then the phoneme recognition can not be done in real time. Although the sub-phoneme (state) recognition result by the neural networks could be given in real time, the accuracy would likely be low.
3. One state per model reduces computational complexity. For the hybrid methods, more states per model will increase the computational complexity for both HMMs and neural networks. In this work, assuming that a neural network with 36 inputs, 100 hidden nodes and 20 outputs is given to estimate posterior probabilities for HMMs with one state per model, if a model has 3 states, then, correspondingly, the neural network will have 60 outputs and the total number of weights will increases from 5756 to 9756.

4.4.4 Training Procedures

Training of the hybrid HMM/ANN ASR system requires estimation of both the parameters of the HMMs and the weights of the neural networks. NNs are usually trained for classification that is required to know target values for the outputs in order to compute the gradient of the cost function. For a large-scale database, hand labeling is not feasible. Considering the NN outputs can be used in the dynamic programming for global decoding, an iterative training procedure, which is called embedded Viterbi learning. An overview of the training procedure is illustrated in Fig 10.

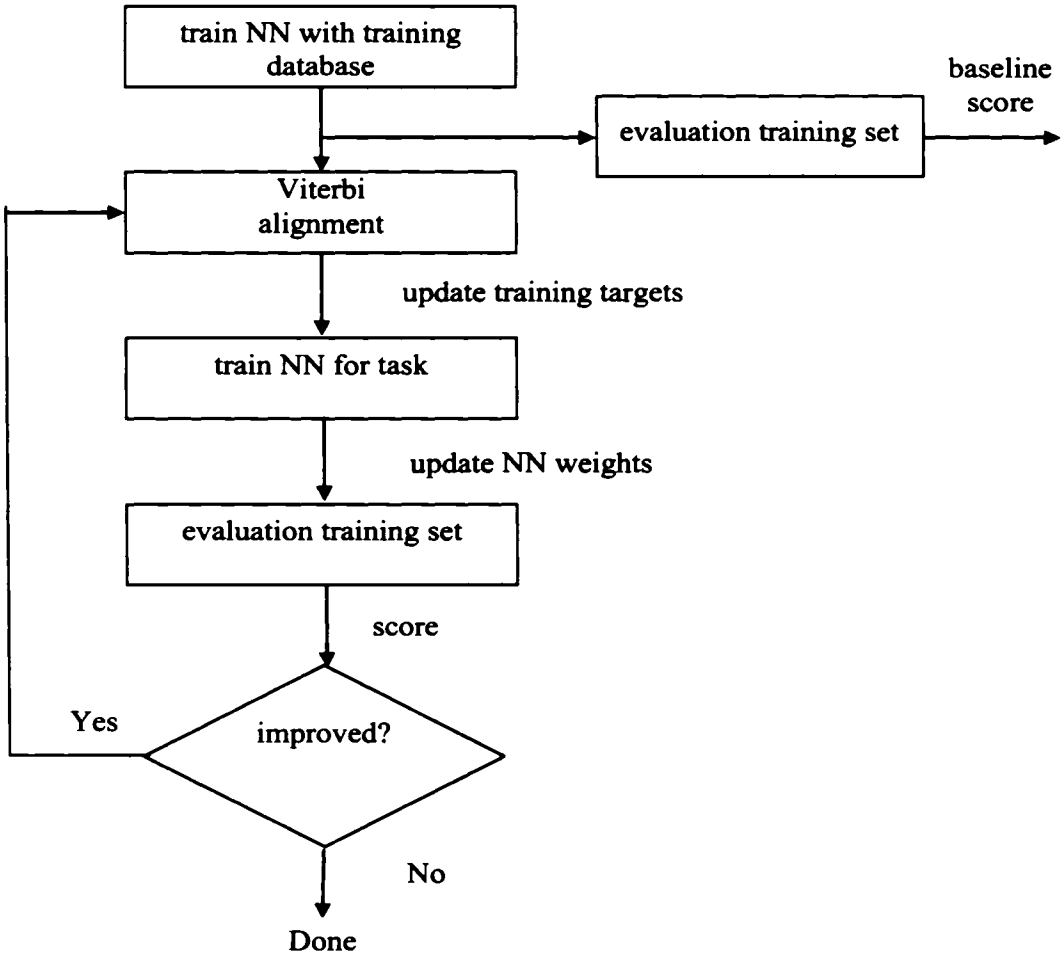

Fig. 10. Embedded Viterbi learning with NNs.

1. Once the model's architecture and database are ready, the neural network is trained with the training set. After the initial training, the currently trained network is used as an HMM posterior probabilities estimator, and the word recognition rate is evaluated and recorded as a baseline score. The Viterbi algorithm is used as the recognizer for this step.
2. Viterbi alignment. In this step, the Viterbi algorithm is used again. But the Viterbi algorithm, while searching for the most probable path for a given observation sequence, is used for alignment. Each speech frame is assigned a new label. The new labels are used as new targets for the neural network training in the next step.

3. Train the neural network with the new targets. Once the new targets for each frame are determined, retrain the neural network with the new targets. In this step, the neural network weights are not assigned from the beginning, but must be updated using the results of the first step.
4. Evaluate the training set again. All training sentences are evaluated again with the same method used in step1 and a new score is obtained. The new score is compared with the baseline score; if the performance is improved, the training procedure is continued and the baseline score is updated with the new score or the training procedure is stopped.

4.5 Experiments and Evaluation

Several experiments were performed based on the methods described in this chapter. Those experiments and the results are presented in this section. The goal was to choose an "optimal" speech recognizer with respect to speech features, the algorithm and parameters of the algorithm. As alluded to previously, both phoneme and word level performances were considered. For all experiments reported in this section, neural network outputs were used as the posterior probability estimators for HMMs. The database was the one described in chapter 2. It contains 14858 CVC short words recorded from male, female and child speakers. All tokens in the database were segmented by a heuristic rule-based segmentation method. The database was divided into two sets: 7000 CVC words were used as the training set and the remaining 7858 files were used as the test set.

The experiments were performed with two types of speech features. One type is MFCC features, which are most typically used in current ASR systems. Each feature

vector has 39 components (12 MFCC parameters plus energy and their first and second derivatives). Another type is DCSC features which have two parameters that can be changed--block length, or number of frames in one block and the number of DCS terms. The feature vector dimension is the product of the number of DCTCs times the number of DCS terms. The other parameters for computing DCSC featured were the same as those described in chapter 3.

As mentioned in the previous section, selection of the number of hidden nodes of neural networks is an important consideration. For most of the experiments reported in this dissertation, neural network had 200 hidden nodes. This number was obtained by performing several experiments. Fig. 11 shows the results of one of these experiments.

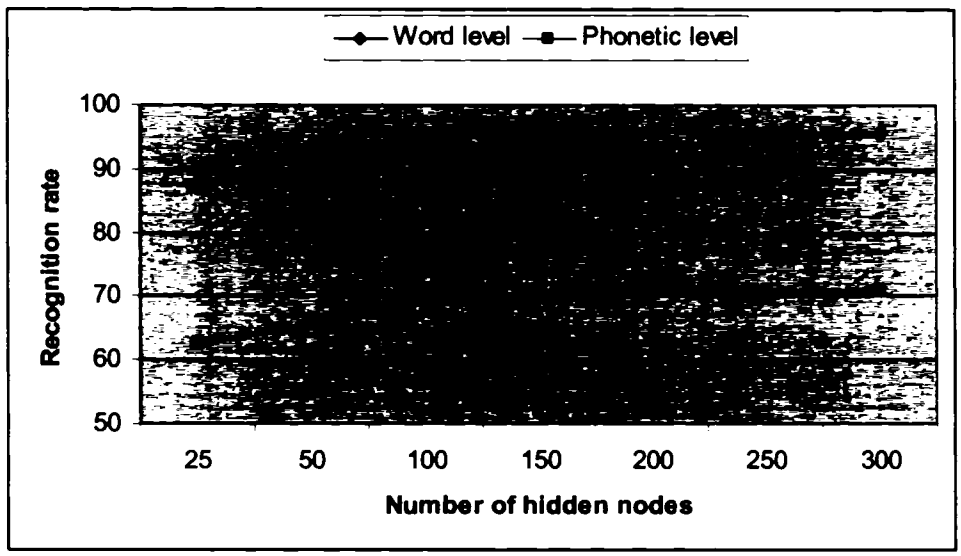

Fig. 11. Word level and phonetic recognition rates.

This experiment was performed with the MLP NN and the CVC database. DCSC features with 3 DCS terms and a block length of 20 frames were used in this experiment. The word recognition results are consistent with phonetic results. For both cases, the best results were obtained when the number of hidden nodes was 200. This number was also

used as the number of hidden nodes for other neural networks. Note, however, that the recognition rate did not change much as the number of hidden nodes was varied from 100 to 300, so presumably the number of hidden nodes could have been any number within this range.

For the hybrid method of combining MLPs and HMMs, the neural networks had 200 hidden nodes and 20 outputs. The numbers of inputs was variable for the different features sets. The structure of the TDNNs for the hybrid method is similar to that for MLPs. The only difference is that the number of input is the product of number of inputs to the MLPs and the number of time delays.

The experiments in this chapter were performed with different methods and different speech features with different settings. The overall experimental results on the test data for this chapter are given in the Table VI.

TABLE VI

EXPERIMENTAL RESULTS (WER) OF HMM/NN METHOD FOR THE CVC DATABASE

Speech feature		**Hybrid method with MLP**		**Hybrid method with TDNN**	
		Phonetic level (%)	**Word level (%)**	**Phonetic level (%)**	**Word level (%)**
MFCC		46.3	13.7	42.2	10.9
DCSC with DCS=3	bk_len=5	41.6	11.3	40.7	8.0
	bk_len=10	38.4	5.1	36.2	4.8
	bk_len=15	34.2	4.2	32.1	3.4
	bk_len=20	31.4	4.1	30.3	3.8
	bk_len=25	30.3	4.4	29.8	4.3
	bk_len=30	31.4	4.9	29.6	4.6
DCSC with DCS=4	bk_len=5	41.8	8.8	42.0	7.5
	bk_len=10	37.8	5.0	35.9	5.4
	bk_len=15	33.9	3.8	32.8	3.9
	bk_len=20	30.8	3.1	30.0	3.4
	bk_len=25	30.2	3.2	28.7	3.7
	bk_len=30	29.6	3.5	27.9	3.3

In Table VI, bk_len indicates block length. The table lists both phonetic and word level recognition results with MFCC features and DCSC features.

4.5.1 DCSC features vs MFCC features

In the previous chapter, the best recognition rate using an HMM/GMM recognizer was obtained with MFCC features. However, for the hybrid method presented in this chapter, all results with DCTC features are better than the results with MFCC features. Based on the experiments summarized in this chapter, the hybrid method with MFCC features does not represent an advantage over HMMs with Gaussian Mixture Models. The lowest WER reported in the previous chapter is 5.1 %, using a HMM/GMM. For the

hybrid method reported in this chapter, the best result with MFCC features is a WER of 10.9 %. In contrast, the hybrid method with DCTC features results in considerably higher recognition accuracy than the HMM/GMM method for DCTC features. As reported in chapter 3, the MFCC features were superior to DCSC features when used with the HMM/GMM algorithm. The highest overall performance was obtained with the hybrid HMM/NN method applied to DCSC features (3.1% WER at the word level, and 28.7% at the phoneme level). Based on these results, DCSC features were chosen as speech feature for the visual speech described in this dissertation.

4.5.2 MLP vs TDNN

The goal of this experiment was to compare the performance of HMM/MLP and HMM/TDNN. Since the previous experiment showed that the DCSC features outperformed the MFCC features, in this experiment, only the results with DCSC features were compared. Additionally, the performance was evaluated for different block lengths. The results are depicted in Fig. 12.

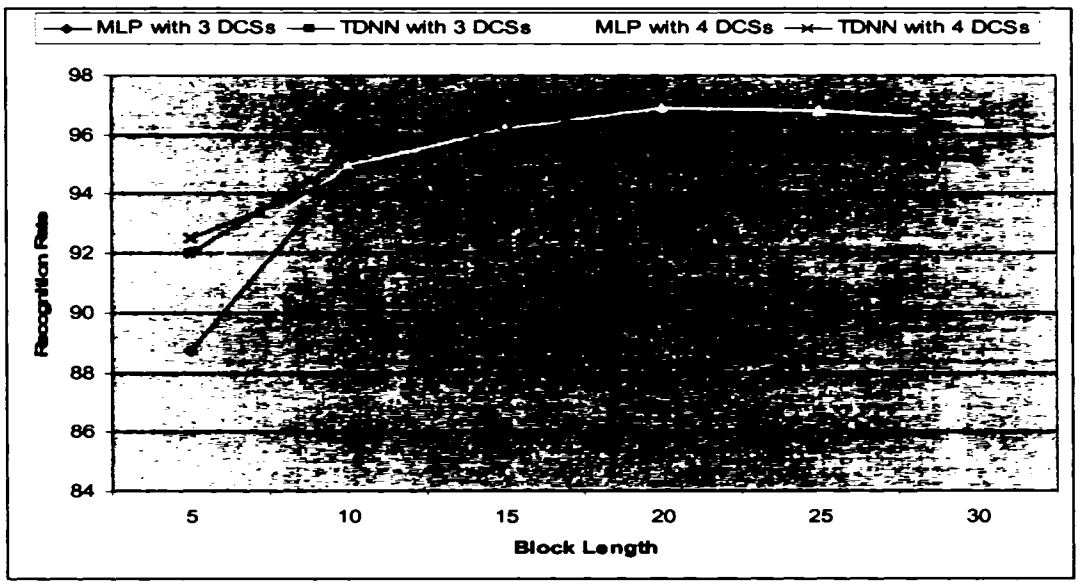

Fig. 12. Word level recognition rates of hybrid methods with MLP and TDNN.

As shown in Fig. 12, for the DCSC features with 3 DCSs, the performance of the hybrid method with TDNNs is better than that of MLPs. However, when the number of DCSs is increased to 4, the performance of the hybrid with TDNNs is very similar to that with the MLP. For a block length is 5, the performance of a TDNN is a little better (8.2% for MLP vs. 7.5% for TDNN). For all other cases, the performance of the MLP is better than for the TDNN. Meanwhile, the MLP has a simpler structure than the TDNN. Therefore, for the real time visual speech display reported in this dissertation, an MLP combined with an HMM is used to build the speech recognizer.

A conclusion can be drawn from the above experiments that the block length affects the performance greatly. The best performance can be obtained when the block length is set to a suitable value. For all methods, the best results were obtained when the block length is 20.

4.5.3 HMM/NN vs HMM/GMM.

In the previous section, the performance of an HMM/NN and HMM/GMM with MFCC features was compared. The results showed that the MFCC features are more suitable for the HMM/GMM. In this experiment, the results were mainly compared with DCTC features. The Table VII lists the results for the two methods with DCSC features.

TABLE VII

COMPARISON OF EXPERIMENTAL RESULTS (WER IN PERCEN) OF HMM/GMM AND HMM/MLP WITH DCSC FEATURES

		Block Length					
		5	**10**	**15**	**20**	**25**	**30**
DCS=3	HMM/GMM	6.9	6.0	5.9	7.3	8.5	9.9
	HMM/MLP	11.3	5.1	4.2	4.1	4.4	4.9
DCS=4	HMM/GMM	7.1	5.8	5.3	6.6	6.4	7.7
	HMM/MLP	8.8	5.0	3.8	3.1	3.2	3.5

Table VII shows the word level recognition results using a HMM/GMM and a HMM/MLP with DCTC features. The performance of the HMM/MLP is better when block lengths are greater than 10.

4.5.4 Phoneme recognition rate vs word recognition rate

In general, phoneme recognition rates are consistent with word recognition rates. The word recognition rates will be improved when phonemes are recognized more accurately. However, this consistency is not always the case. In Table VI, the lowest phoneme error rate is 27.9% when the method is HMM/TDNN and DCTC features with 4 DCS terms and a block length of 30. The corresponding WER is 3.3%. This result is higher than 3.1%, whose corresponding phoneme error rate is 30.8% and obtained with 4 DCS terms and a block length of 20.

4.6 Chapter Conclusion

In this chapter, methods of combining neural networks and HMMs were discussed. The methods of using neural network outputs as posterior probability estimators of HMMs were emphasized.

The results of hybrid methods with two kinds of speech features – MFCC features and DCTC features were compared. Although MFCC feature are merely used in current speech recognition applications, in this dissertation, the recognition results obtained using the hybrid method with DCTC features was superior overall.

Several experiments were performed using an MLP or TDNN as a posterior probability estimator for HMMs. The results showed that the word recognition rate were similar for these two neural networks. However, the architecture of the MLP is simpler than that of the TDNN and thus requires fewer computations, which is an important factor for a real time speech articulation training aid.

CHAPTER V

NORMALIZATION OF FEATURES FOR SPEECH RECOGNTION

5.1 Introduction

In the pervious chapter, an HMM/NN hybrid recognizer was presented for CVC speech recognition. In that method, neural network outputs acted as posterior probability estimators for HMMs. In this chapter, another hybrid HMM/NN method is presented which uses neural network outputs as new speech features.

As mentioned in the previous chapter, one unreasonable assumption underlying HMMs for speech applications is that acoustic features are uncorrelated. This assumption provides the basis for using diagonal covariance matrices instead of full covariance matrices in Gaussian mixture models to reduce computational complexity. Another problem with the HMM/GMM method is the non-discriminative training which thus results in poor discrimination at the phonetic level.

The hybrid method discussed in the previous chapter can overcome these shortcomings. However, this method also has limitations:

1. The HMMs used, in order to make computations reasonable, were too simple.

   The HMMs used for the method presented in chapter 4 were basic monophone models. That is, only one model was created for each phoneme and only one state was used for each model. However, the speech process is more continuous. The disadvantage of monophone models is that the transition between two adjacent phonemes is not smooth; thus the final word level recognition rate is degraded due to modeling inaccuracy. More complex models may be used to solve this problem. For example, each phoneme model could have 3 states.

Although the models are still monophone models, the transitions within each model would better model the actual speech process. However, these additional states increase computational complexity. A model with 3 states means the number neural network outputs would increase by a factor of 3, relative to a model with 1 state. More fundamentally, the training of the neural networks for each state is problematic in the sense that much of the discriminative power of the neural networks is "wasted" discriminating states within a phone versus states between phones. In some pilot tests, increased performance could not be obtained at the phonetic level with additional states per phone for this hybrid approach.

2. The accuracy of neural network performance affects HMM results greatly.

In the previous chapter, a brief proof is given on why the outputs can be interpreted as estimates of a posteriori probabilities. According to the proof, a requirement is that the neural network system must be sufficiently complex to have a good approximation to the mapping function between input features and the output class. However, in practice, this requirement is hard to satisfy due to the variability of speech signals. For example, the best phoneme recognition rate obtained from experiments reported in the previous chapter was 72.1%. That means many blocks were classified incorrectly by the neural network and thus the mapping function between the input and the output class was not correct. Although neural network system complexity could be increased by using more hidden nodes, that would increase computational complexity and might result in poor generalization.

The most significant factor affecting speech recognition performance is variation in the signal characteristics from trial to trial (intersession variability and variability over time). It is important for automatic speech recognition systems to accommodate these variations. One of the techniques that accommodate speech feature variations is called normalization. This technique can project the speech features from their original speech to a new feature space. The intention is that, in the new feature space, the unwanted sources of variability are reduced. Normalization methods can also be used to reduce correlations between features, thus making the new feature space more suitable for recognizers that assume independence of features.

Using neural networks to map speech features from one space to anther space has been studied. In 0, the neural network plays the role of a feature extractor for a standard continuous density hidden Markov model (CDHMM). In this chapter, the neural network outputs are used as new speech feature vectors, and then HMMs are trained with these new speech features.

5.2 Method description

The goal of normalization method is to remove variation in the speech features. That is, if $\boldsymbol{x}$ is any feature vector in the original feature space, then a transformation **T** is expected, such that the transformed vector in a new features space $\boldsymbol{y}$ is:

$$y = T(x) \quad \forall(x, y) \tag{5.1}$$

where y is uncorrelated. Additionally, the variability in the original features should be reduced while preserving phonetic discriminability.

Correlation in the feature space can be measured by the relative magnitude of the off-diagonal components of the covariance matrix. To support the claim that the feature space is indeed correlated, and that the 'extent' of correlation varies with environmental conditions, the following metric was adopted to empirically evaluate features for different conditions [73]:

$$k = \frac{\left\| R - R^{diag} \right\|}{\left\| R \right\|} \tag{5.2}$$

where R is the covariance matrix and R^{diag} is the matrix obtained by deleting all the off-diagonal elements from R.

k is hence a measure directly proportional to the extent of non-diagonal behavior exhibited by R. For a strictly diagonal matrix, $k = 0$ and for a matrix with zeros on the diagonal and non-zero off-diagonals, $k = 1$.

The neural networks have been shown to work as nonlinear mappings from one feature space to another feature space [84]. If the neural network training targets are set appropriately, then the transformed feature vector will be uncorrelated or less correlated. In this dissertation, the same neural network training targets are used as the neural network classifiers in the previous chapter; that is, for the input vectors in one category, their corresponding output node is set to 1 whereas other nodes are set to 0. The target vectors for different categories are orthogonal. In the ideal case, the covariance matrix of transformed features should be an identity matrix or close to an identify matrix. Then the k in formula 5.2 is close to 0. That means the correlation of normalized features is low.

Once the neural networks are trained, the speech features are transformed with the neural network and become normalized features. Then HMMs are built and trained with

normalized speech features. Compared to original speech features, the normalized features have the following advantages:

1. The normalized features are more uncorrelated and compact.

   For the method presented in this chapter, the number of normalized features is equal to the number of phonetic categories. The neural network is trained to attempt to map each set of speech features to its phonetic category. Thus the target vector for training always consists of a "1" in a position corresponding to the index of the phoneme for that feature vector, and zeros in all other positions.

   Then all speech features will be normalized with the neural network. The normalized features between different categories tend to become uncorrelated, as illustrated by the experimental results presented below. For the CVC database used in this dissertation, the number of phoneme categories is 20, which is less than the dimension of the original speech features (more than 36) used. Thus the speech features are more compact after normalization.

2. The computational complexity of HMMs will be reduced.

   The GMMs were used to calculate posterior probabilities of HMMs for normalized features. Since the normalized features are more uncorrelated and compact, then fewer parameters are required for HMM training. Thus, the computational complexity is reduced.

3. HMM results based on the new speech features are more robust, as illustrated by the experimental evaluation reported in the next section.

5.3. Experiments and Evaluation

The same database was used as for the previous two chapters. The database has 14858 CVC short word utterances. The first 7000 utterances were used as training data and the remained 7858 utterances were used as test data. The results given in chapter 4 showed that the performance with DCSC features is better than that with MFCC features. In this chapter, all experiments were performed using DCSC features with different block lengths.

The neural network in this chapter was used to normalize original speech features. To compare performance, the same neural network architecture was used as for the experiments reported in the previous chapter. The neural network had 200 hidden nodes and 20 output nodes. The number of input nodes was varied with different block lengths when computing DCSC features

5.3.1 The comparison of the correlation of normalization features and original features.

In this experiment, the correlations of the original speech features and normalized feature were evaluated and compared. Correlation metric k was calculated with (5.2). Table VIII shows the correlation metric for each phonetic category and the overall correlations over all phonemes.

TABLE VIII

THE CORRELATION MATRIX FOR EACH CATEGORY AND THE AVERAGE CORRELATION

Phone	Original features	Normalization features	Phone	Original features	Normalization features
'b'	0.7476	0.3148	'eh'	0.7540	0.1760
'd'	0.8636	0.3922	'er'	0.7181	0.1925
'g'	0.8616	0.3281	'ey'	0.7988	0.2304
'k'	0.8479	0.4021	'ih'	0.7581	0.2299
'p'	0.8670	0.4092	'iy'	0.7990	0.2040
't'	0.8439	0.2810	'ow'	0.7739	0.1958
'aa'	0.7401	0.2193	'oy'	0.7028	0.2150
'ae'	0.6992	0.2617	'uh'	0.7946	0.1612
'ah'	0.8034	0.1669	'uw'	0.6718	0.2125
'ao'	0.6991	0.2071	'sil'	0.8589	0.2802
Overall	0.78017	0.253995			

All values in Table VIII are from the test data set. The original speech features were 36-dimensional DCSC features (12 DCTC terms and 3 DCS terms) with a block length of 20. In Table VIII, the correlations of normalized features are smaller than correlations of original features for each phoneme category. Thus, the neural network does reduce feature correlation. For the original features, the correlations of features are high for both consonants and vowels. However, the correlations of vowel normalization features are less than those of consonant normalization features. (0.3546 for consonants and 0.2109 for vowels). It shows that the neural network works better for removing feature correlations of vowels than for consonants.

5.3.2 Word recognition performance

Several experiments were performed with both normalized speech features and standard HMMs. Two kinds of HMMs (monophone models and triphone models) were used for the experiments. Note, however, that the HMMs implemented for these normalized features had fewer parameters than the HMMs implemented for the work presented in the previous chapter. For the results given in chapter 3, the dimension of the speech vector was 36 (12 DCTC terms over frequency, each represented with a 3-term DCS expansion over time) or 48 (12 DCTC terms over frequency, each represented with a 4-term DCS expansion over time). For the experiments reported in this chapter, the normalized speech features had a dimension of 20, irregardless of the dimension of the original speech features. For the HMM/GMM experiments reported in the previous chapter, 5 Gaussian mixtures were used for each model. For the experiments with normalized features, reported in this chapter, 3 Gaussian mixtures were used for each model.

The HTK toolkit was again used for HMM training. The same training strategy was used as that used for the experiments reported in chapter 3. The training process began with monophone models. Then the triphone models were created by copying the parameters of trained monophone models with context. Baum-Welch and Viterbi algorithms were used in the training and recognition process.

The final results are reported in terms of percent accuracy for the test data. Table IX shows the word level recognition rates of standard HMMs, hybrid of HMM/NN as described in chapter 4 and the method of HMMs with normalized features.

TABLE IX

WER (IN PERCENT) FOR AN HMM RECOGNIZER USING NORMALIZED FEATURES

DCS	Block length	Monophone		Triphone	
		Training	Test	Training	Test
3	5	6.1	9.9	2.8	5.7
	10	3.3	5.8	2.1	5.1
	15	2.4	5.6	1.4	5.7
	20	2.2	5.8	1.3	5.2
	25	2.0	5.3	1.2	4.9
	30	1.8	6.0	1.0	4.0
4	5	5.6	8.5	3.0	5.8
	10	2.9	6.1	1.6	4.2
	15	2.2	4.8	1.0	2.7
	20	1.4	4.0	0.9	2.7
	25	1.3	4.0	0.8	2.6
	30	1.1	4.8	0.7	3.0

In Table IX, the number of DCSs and the block length refer to the original DCSC features. As for the results given in chapter 3, the results using triphone models are always better than those obtained with monophone models. Experiments were conducted with different block lengths and various numbers of DCS terms. The best test result obtained was 2.6%, using triphone models with 4 DCS terms and a block length of 25.

To compare the performance of HMMs with normalized features with performance obtained with the other two methods, only the test recognition rate for the three cases are depicted.

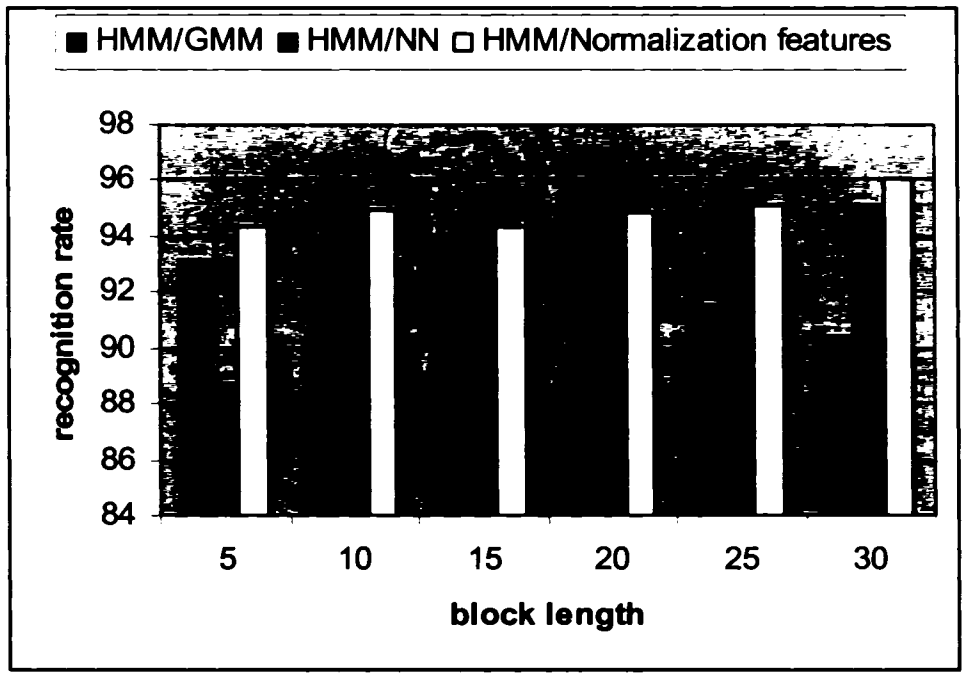

Fig. 13. Comparison of recognition rate for HMM/GMM, HMM/NN and HMM/ normalization features with DCS = 3.

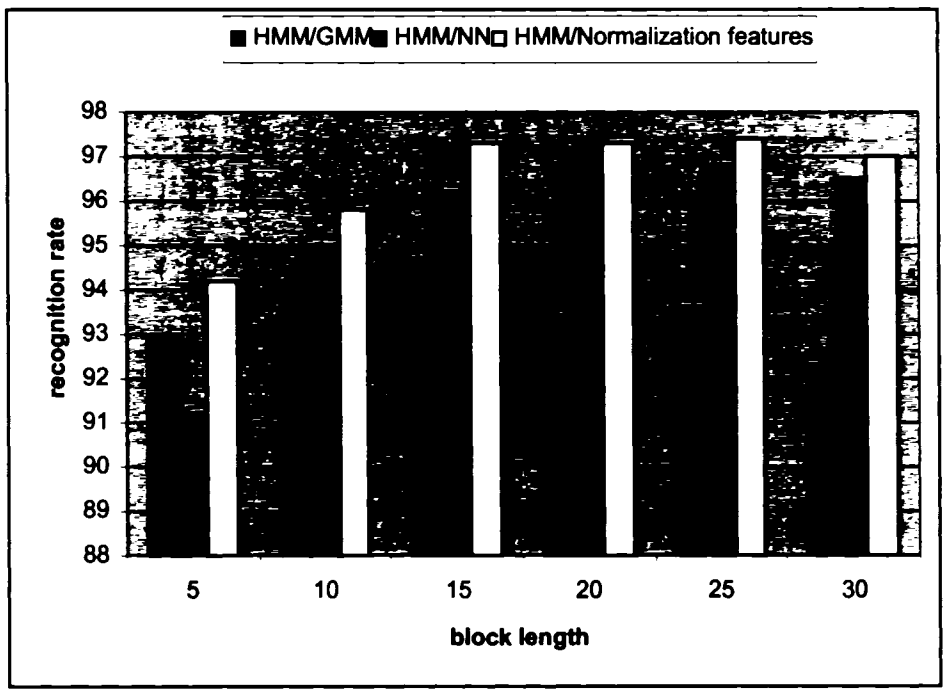

Fig. 14. Comparison of recognition rate for HMM/GMM, HMM/NN and HMM/ normalization features with DCS = 4.

Fig. 13 shows the recognition rate results for HMM/GMM, HMM/NN and HMM/Normalization features with 3 DCS terms. For most of the cases reported in Fig 13, the results for the HMM/NN are better than for the other two methods. However, the best result overall (96.0% and WER is 4.0%) was obtained using HMM/ Normalization features with a block length of 30.

Fig 14 shows the results for the three methods with 4 DCS terms for the feature space. The use of HMMs with normalized features was superior to the other two methods. The best overall result (97.4% and WER is 2.6%) was obtained with HMM/Normalization features with a block length of 25.

5.4 Conclusion

In this chapter, a new hybrid method of HMMs and neural networks was presented. The neural networks presented in the chapter transforms the original speech feature to normalized speech features. Compared with original speech features, the normalized features have less correlation and are more suitable for HMM assumptions. Also, in the HMM training process, fewer parameters are needed with normalized features. The experiments showed that the normalized features have less correlation than the original features and give better results, compared to hybrid methods with original features.

CHAPTER VI

CVC DISPLAY

6.1 Introduction

The objective of the CVC display system is to visually present indicators of pronunciation "correctness" at the phone and word level, in response to short words produced by a speaker. The visual speech display should present a maximum amount of speech information in an integrated, continuous, easy-to-interpret, and appealing form such that small changes in pronunciation result in immediate small variations in the display pattern. Besides the recognition results, the system also displays short term energy and pitch tracking information. The details are described in the following sections of this chapter.

In order that this speech training aid can be widely used, the system uses only a "standard" PC with a microphone and sound card, thus making development and system costs inexpensive. The system was developed for the windows operating system with C++ object oriented programming.

The system includes three modules corresponding to speech signal processing, speech recognition, and graphical user interfaces and displays. The speech signal processing module collects raw speech acoustic waveform to a continuous "segment" buffer, detects endpoints of the speech signals and converts the received raw speech signal to speech feature vectors. In this system, speech signals are converted to DCSC features. Speech recognition is the kernel module. The system's performance is based on the performance of the speech recognition. Since this system is for articulation speech

training for people with hearing impairments, it will only indicate "correct" responses if the users' voice inputs are pronounced correctly. Ideally users will know the quality of their pronunciations and how to correct their incorrect pronunciations. Thus the performance of the speech recognizer plays an important role for the system's validity. The speech recognition algorithms discussed in previous chapters were used to build the speech recognizer. Two levels of recognition results are provided to users: at the phonetic level and at the word level. Visual displays are also very important aspects for the system since users communicate with the system through these displays. The displays are intended to be friendly, easy-to-interpret and present a maximum amount of speech information to users. Two types of displays are provide for users; one is a "flow mode" display, which depicts phonetic information, pitch and the energy contour flowing across a computer screen from right to left. Another display is a "block" display, which simply gives word recognition results by highlighting blocks on a screen. Besides these two displays, some other displays also were implemented, primarily to illustrate intermediate processing steps. These include a waveform display, a short-time frequency spectrum display, two bargraph displays representing speech features and neural network outputs, and a spectrograph (time-intensity-frequency) display. These displays can be used for system diagnostic purposes, and can also be used by a speech language pathologist as part of a speech training program.

This display as well as other displays can work under two modes: real-time mode and non-real-time mode. For the real time mode, the system responds to the input acoustic signal with the least time delay. Then users receive feedback about pronunciation correctness immediately. In contrast, for the non-real-time mode, the

signals are saved in a buffer and users are able to analyze the signal at any selected time moment, but only after some time delay.

6.2 Signal processing

A very important consideration in the overall implementation of the display is to minimize delays between input speech and display changes—to make the overall system as "real-time" as possible. In order to understand how the "real-time" aspect of the display is implemented, it is useful to describe the signal acquisition/processing/ display steps from a data delay point of view. As to signal acquisition, the operating system interacts with the sound card to continuously acquire contiguous sections of data ("segments") from the audio data stream, using a double buffering approach for processing. The basic data acquisition was developed with operating system services, rather than directly with the hardware, and therefore the overall CVC system is able to work with most commonly available Windows-compatible sound cards. The data management for this signal processing can be explained in terms of three levels of buffering, as illustrated in Fig. 15.

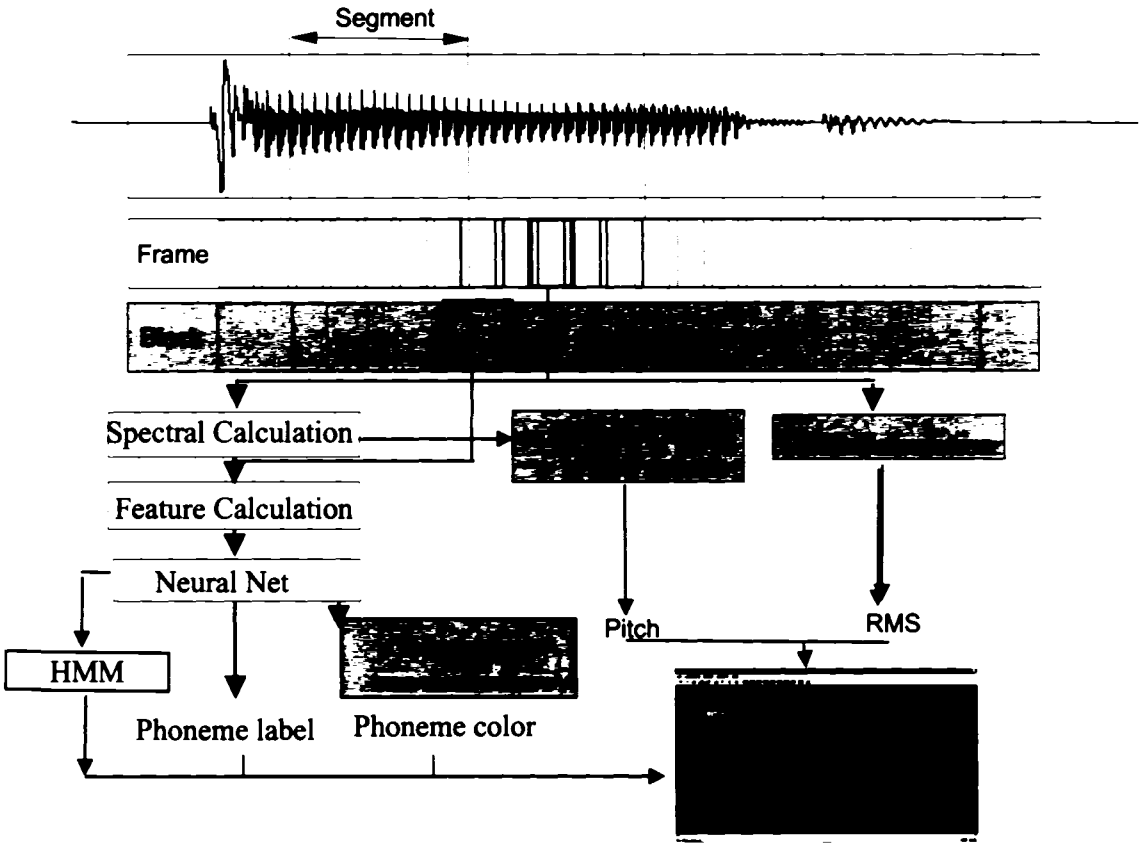

Fig. 15. Overview of signal processing for CVC display.

First, the non-overlapping but continuous "segment" buffers form an interface between the data acquisition subsystem, typically a sound card, and the training aid system. The first step of processing is based on overlapping frames of data — thus frames form the second level of buffering. The third and final level of buffering is referred to as "blocks", with each block consisting of an integer number of frames, with block spacing also consisting of an integer number of frames.

There are three other signal processing parts implemented in the existing system. The short term energy calculation, the pitch tracking and the transformation of the phoneme position in the L*u*v* color space to a color in the RGB color space. The details of these parts are discussed in the following sections.

The original intention was to use the results of the existing (from VATA) spectral transformation for both the feature calculation and the pitch tracking. But tests showed that the phoneme recognition, based on the features, is better using small frames, whereas more robust pitch tracking was achieved using longer frames. Because of this observation the framing of the segments and the spectral calculation are performed twice.

6.2.1 Energy Contour

The energy or loudness contour is calculated using Root Mean Square (RMS) averaging. The basic calculation is to take the square root of the sum of all squared signal values in a frame. One RMS value is obtained per frame. The following formula describes this computation [74]; x_i is the ith sample inside the frame with a total width of W speech samples.

$$RMS = \sqrt{\frac{1}{W}\sum_{i=0}^{W-1} x_i^2}\ . \tag{6.1}$$

To achieve a perceptual scale, all RMS values are converted in decibel (dB) using the base 10 logarithm. The measurement is relative to the reference energy of a pure sinusoidal function with maximum amplitude. In the case of 16-bit resolution sampling the calculation is performed as shown in the next equation:

$$RMS_{db} = 20\log_{10}\frac{RMS}{(2^{15}/\sqrt{2})}. \tag{6.2}$$

Using this conversion a dynamic range from 0 to -90 dB (for silence) is obtained.

6.2.2 A Simple Pitch Tracking Algorithm

In order to display the pitch, the original intention was to incorporate the pitch tracking from the YAPT tracker [75], but instead a simplified version of the tracking was chosen. In particular, the pitch track is obtained by pick-peaking in the low frequency part of a spectrogram. The actual algorithm locates the first peak by scanning through the frequency components, starting at the lowest frequency. As soon as the difference between a component and its predecessor is positive, the first negative difference marks a peak. Fig. 16 illustrates this process.

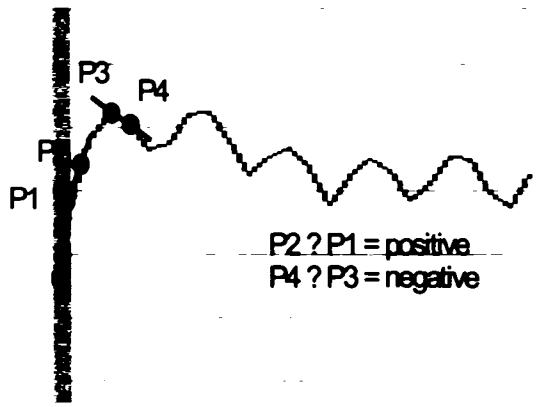

Fig. 16. First peak detection for pitch tracking.

In the actual system a spectrogram from 50 to 400 Hz is used for the pitch tracking. The total spectrogram ranges from 50 to 3000 Hz and is computed from a frame of 30 ms, using a sampling rate of 11025 Hz and a 512 point FFT (Fast Fourier Transform). Best results were achieved without additional time or frequency smoothing.

Additionally, the spectrum also serves for calculating an energy ratio useful for discriminating voiced and unvoiced speech parts. The calculation is done using the following formula:

$$ratio = \frac{\sum_{i=50\,Hz}^{400\,Hz} component_i}{\sum_{i=50\,Hz}^{3000\,Hz} component_i}. \quad (6.3)$$

The first sum is computed using the frequency components in the range from 50 to 400 Hz, whereas the second sum uses components from 50 to 3000 Hz. A threshold is used to distinguish between voiced and unvoiced parts. Empirically [76], it was determined that if the ratio is greater than 1.1 that section of speech is likely to be voiced. Furthermore, sections with an RMS value smaller than -40 dB are considered as silence. However, neither of these discriminations based on simple thresholds (voiced versus unvoiced; speech versus silence) appeared to be particularly accurate. Additional work is needed if more accuracy is required. This point was not pursued in the course of this research, since the simple techniques just described were suitable for the purposes of the visual speech display.

The recognizer decisions are made from all the block output parameters. Display updates are made once per each new segment, but using all the block-based recognizer decisions which occur in that segment. Each of the processing and recognition modules maintains internal delay memories that hold all previous data and intermediate calculations that are needed. Thus the "frame' and 'block' level buffers can be asynchronous with respect to the segment buffers. Typical values for the various buffers are 60 ms for the segment buffer, 20 ms frames with a frame spacing of 10 ms, and blocks consisting of 20 frames with a spacing of 1 frame between blocks. The time delay between input speech and display changes is on the order of a maximum value of one segment time plus one block length time.

Another important aspect of the data management is automatic endpoint detection,

both for the beginning and end of the utterance. The endpoint algorithm is based on energy measures and threshold logic. The thresholds are based on a sample of the background acoustic noise in the environment. Since the signal acquisition is always in operation, using the double buffering approach mentioned above, the effective signal processing can extend both before the beginning start point and after the final end point. In the display, the effective signal processing is synchronized with the endpoints, but extends 5 segments (about 300 ms) in either direction from the detected endpoints. With this approach the very important initial and final bursts of stop consonants are less likely to be missed.

The actual signal processing consists of mid-frequency pre-emphasis, windowing, spectral magnitude calculations and speech feature computation. Based on the experimental results given in chapter 4 and chapter 5, DCSC features were chosen for this system. First, DCTCs are computed for each spectral frame and then DCSs are calculated over the DCTCs in each block, and scaled. In the final step, the DCTC's are block-encoded with a sliding overlapping block using another cosine transform over time that is used to compactly represent the trajectory of each DCTC. The cosine basis vectors in this second transform are also modified so that the temporal resolution is better near the middle portion of each block relative to the endpoints. DCSCs are the coefficients of this second transform. In this system, the block length was set to 25 and 4 DCS terms were used.

6. 3 Speech Recognizer

In chapter 4 and chapter 5, two methods of combining neural networks and HMMs for word level speech recognition were represented. The first method uses neural

networks as the posterior probability estimators for HMMs. In the second method, the neural network transforms the original speech feature to normalized speech features. Then HMMs are trained with normalized features. Based on experiments, the best performance was obtained with the second method. For both methods, the neural network outputs were also used as phonetic recognition results.

Currently, the speech recognizer in this system is based on the first method. That is, the original speech features (DCSCs in this system) are input into a neural network, the network outputs are used as posterior probabilities for HMMs as well as phonetic results.

The reason of using the first method is based on two considerations: 1. The HMMs in this method are simple and easy to implement. Monophone models are used to create HMMs and each model has only one state (except for initial and final states.) For the second method, better performance was obtained with triphone models using at least three states per model. The observation probabilities are computed with Gaussian mixture models obtained from normalization features. Additionally, until the time of this writing, the training and testing of the second method were performed with HTK, which is complex to recode for the real time system. 2. Although the performance of the first method is lower than for the second method, the difference in results is quite small. The best test result for the first method is 96.9% which the best result of the second method is 97.4%. Thus the performance obtained with the first method is quite acceptable.

The neural network has three layers with 48 nodes in the input layer, 200 nodes in the hidden layer and 20 nodes in the output layer. The number of input nodes and hidden were determined by the experimental results in chapter 4 to achieve best performance.

There are 20 HMMs in the system and each model corresponds to a phonetic category. Not including the initial and final states, each model has only one state.

6.4 Graphical Outputs

Several graphical outputs were developed for the system. They are: a waveform display, a short time frequency display, a flow-mode spectrograph display, a feature bargraph display, a bargraph display indicating phoneme recognition result, a flow-mode phoneme display and a block display indicting word recognition result. Each display corresponds to a different signal processing stage, as illustrated in Fig. 17

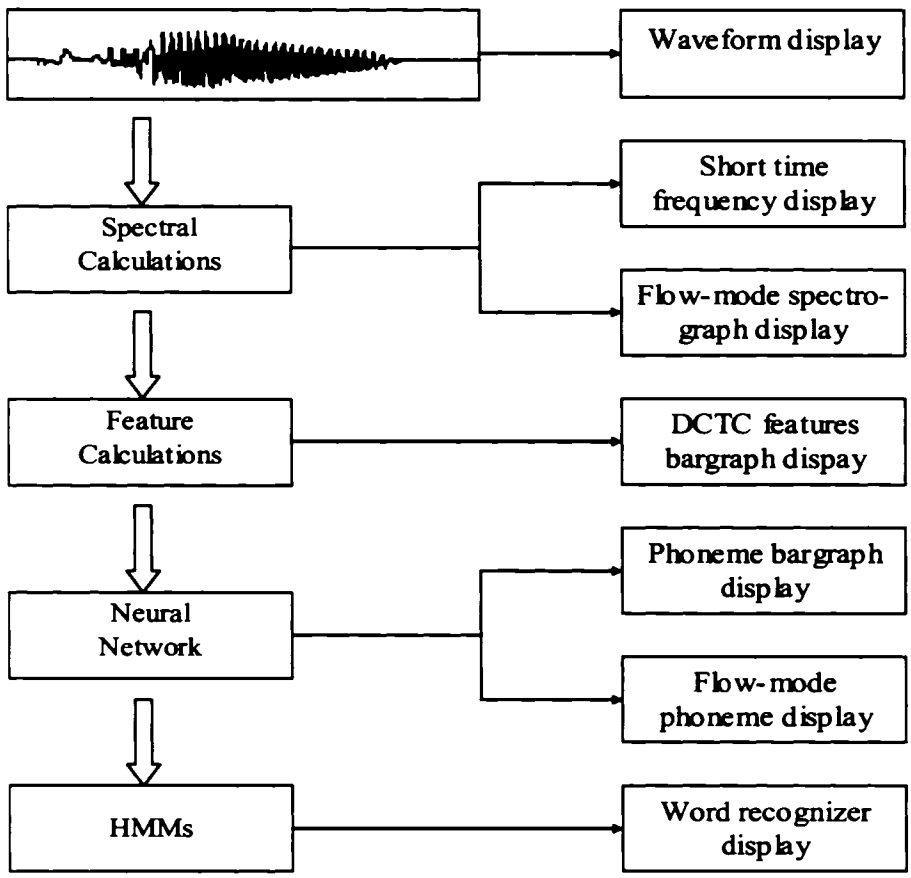

Fig. 17. CVC display signal processing stages and corresponding graphical outputs.

The primary visual display is the flow-mode phoneme display since most of functions were integrated into it and it was intended as the primary display for articulation trainings. Other displays can be used as speech analysis tools. The flow-mode phoneme display is discussed in some detail and the other displays are summarized briefly, all in the following sections.

6.4.1 The flow-mode phoneme display

This display was designed to depict phonetic information, pitch and the energy contour as well as the acoustic speech information in flow-mode. Color, size and texture, and even text labels are used to convey information. Of the vast number of potential methods for creating such a display, the following format was adopted:

1. The basic form of the display is a two-panel display, with the acoustic speech information (primarily envelope) displayed in one panel, and a color coded CVC (consonant vowel consonant) displayed in the lower panel. Alternatively the upper panel can be used to depict a color-coded CVC from a "teacher" and the lower panel used to depict the CVC production of the student.

2. The short term energy (i.e., loudness), is used to control the contour of the CVC display. A 20-output neural network is used to get the highest ranking phones and a weighting of the outputs to determine the phonetic position inside the so-called L*u*v* color space. The position determines the color of each block in the display. Simultaneously, the phonetic transcription labels are written below each band of color. The phonetic transcriptions are obtained with HMM alignment

3. Additionally, the pitch information is also superimposed as a single line over the color CVC display. Furthermore unvoiced portions of the display are depicted as

a textured color, whereas voiced regions are depicted as solid colors. Portions judged to be silent are depicted as a gray patterned color.

4. The graphics makes use of the whole window space available and is scalable if the program window size is changed. A parameter can be set to control the window width (time). If the parameter is chosen as too small for the signal to display, it will adjust itself to display all the data.

An example of the flow-mode phoneme display is illustrated as Fig. 18

Fig. 18. A CVC display in response to the word "Bird" by a male speaker.

In Fig.18, the top panel displays the speech acoustic time waveform. The phoneme recognition results for each block are displayed in the lower panel in the corresponding position with different colors and textures. Two sets of labels are also displayed. The top set of labels indicate the highest ranking phone recognized by the neural network for each block and the lower set of labels indicate the phoneme

recognition results by the HMM/NN. The computed loudness is used to control the height of each block of color in the lower panel. Thus the contour of the lower color pattern indicates the loudness of the utterance over time.

The pitch contour is indicated by the superimposed yellow lines shown in the lower panel. An inspection of several utterances showed that the pitch display is reasonably accurate in the clearly voice positions of the CVCs, but is often in error near the beginning and end of each word. A detailed investigation of the pitch tracking was beyond the scope of this study.

A very important consideration is how to display the results of the recognized phone by the automatic speech recognizer. If a neural network is used as a classifier, typically, each input vector is assigned to the category with the highest output value, and only that category label is then used. However, for the application of using ASR methods for speech articulation training, the users need to be given an indication not only of the correctness of their pronunciation, but also information that could be used to guide the user to correct the incorrect productions. Therefore, the feedback of the system should contain both of the classification results and the distance of the recognized phone to the intended pronunciation.

In this work, the ASR results for the pronunciation of each phoneme are represented with colors and textures. Given that color is intended as a major component for the display of phonetic information, care was taken to map phones to a perceptual color space instead of the non-uniform RGB color space used to display colors for most computer systems. In particular the phonemes are mapped to the L*u*v* color space to

define a color. For display purpose the chosen color is then converted back to the RGB color space.

The L*u*v* [76], sometimes also written as CIELUV is a perceptual color spaces CIE agreed on. CIE (International Commission on Illumination) establishes standard systems for specifying color in terms of human perception and models form the basis for most quantitative color measurement. The intention is to create a perceptually uniform space; that is, equal distances in the color space should be perceived as equally different colors by humans, for any location in the space. The L*u*v* color system is more perceptually uniform than the RGB color space. In the L*u*v* system, L* stands for lightness and u* and v* are chrominance components. The L*u*v* color space and RGB color space are transformable through another color space --- XYZ color space. The details of transformations are described in [77]. Fig. 19 shows the L*u*v* space with the color gamut of the RGB system.

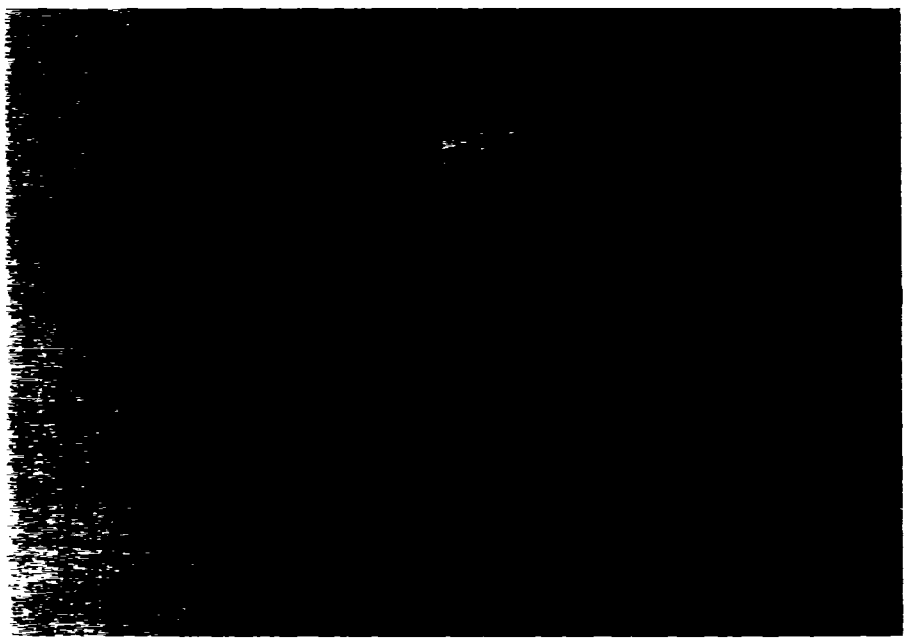

Fig. 19. L*u*v* color space with RGB and standard observer gamut.

In Fig. 19, a color gamut is the area enclosed by a color space in three dimensions. The black line shows the edges of the gamut of the standard observer – the area encloses all colors a human can perceive.

For the visual speech display, the L* component was set to a fixed value since it only stands for lightness but does not contain chrominance. It was found that the space with L* = 62 is a good choice, because it is one of the planes with the biggest extents and the shape of the RGB gamut is similar to the vowel space used in the ellipse display for VATA. U* and v* components were used to represent the mapping of neural network outputs. That is, the 20-dimensional neural network output vector was mapped to a 2 dimensional vector (as explained below) indicting u* and v* components in the L*u*v* color space. Then, the obtained position in the L*u*v* color space is transformed to the corresponding RGB color for computer display.

An important consideration is how to determine the transformation function which maps the 20-dimensional neural network output vectors to 2-dimensional vectors. A bottleneck neural network was used to transform neural network outputs to 2-dimension space. The architecture of bottleneck neural network is illustrated as Fig. 20.

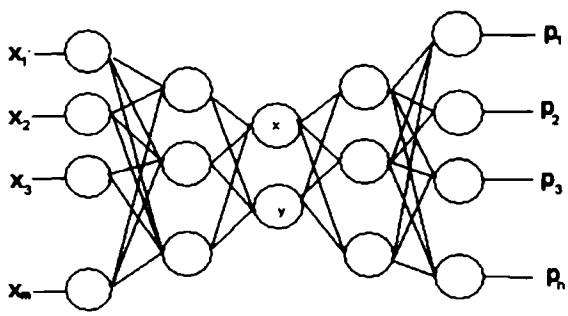

Fig. 20. Architecture of bottleneck neural network.

The neural network has one input layer with 36 nodes, 3 hidden layers with 150, 2, 150 nodes respectively and one output layer with 20 nodes. Each node in the output layer represents a phone category. After training, when a feature vector is input to the network, the outputs from the second hidden layer, which has 2 nodes, are used directly for coordinates in an L*u*v* color space. Fig. 21 illustrates the positions of each phone in a 2 dimensional space trained by the bottleneck neural network.

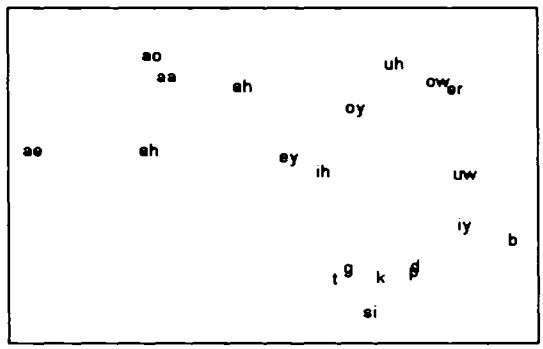

Fig. 21. Locations of each phone in a 2-d space.

In Fig. 21 the positions of similar phones are close to each other. Note the positions are similar to what might be expected from a multi-dimensional scaling of these phones. The whole process of mapping from a speech feature vector to a 2-d space is illustrated as Fig. 22

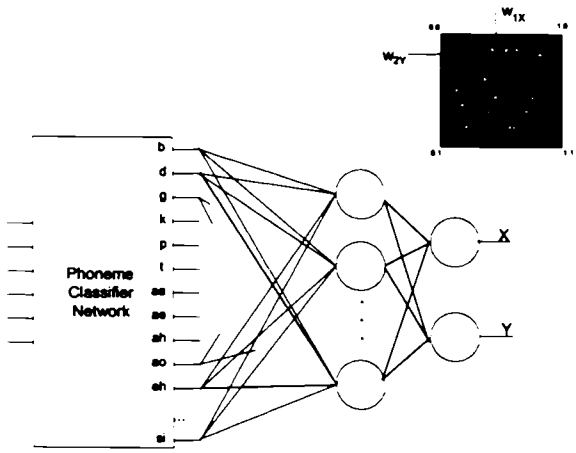

Fig. 22. Mapping speech features to a 2-d space.

6.4.2 The waveform display

This displays the acoustic waveform in the time domain. In addition to the voiced speech signal (as determined by the endpoint detection described previously), the preceding 500 ms before the onset and following 500 ms, after the final endpoint, are displayed. Typically these 500 ms sections are mainly silence. An example is illustrated as Fig. 23.

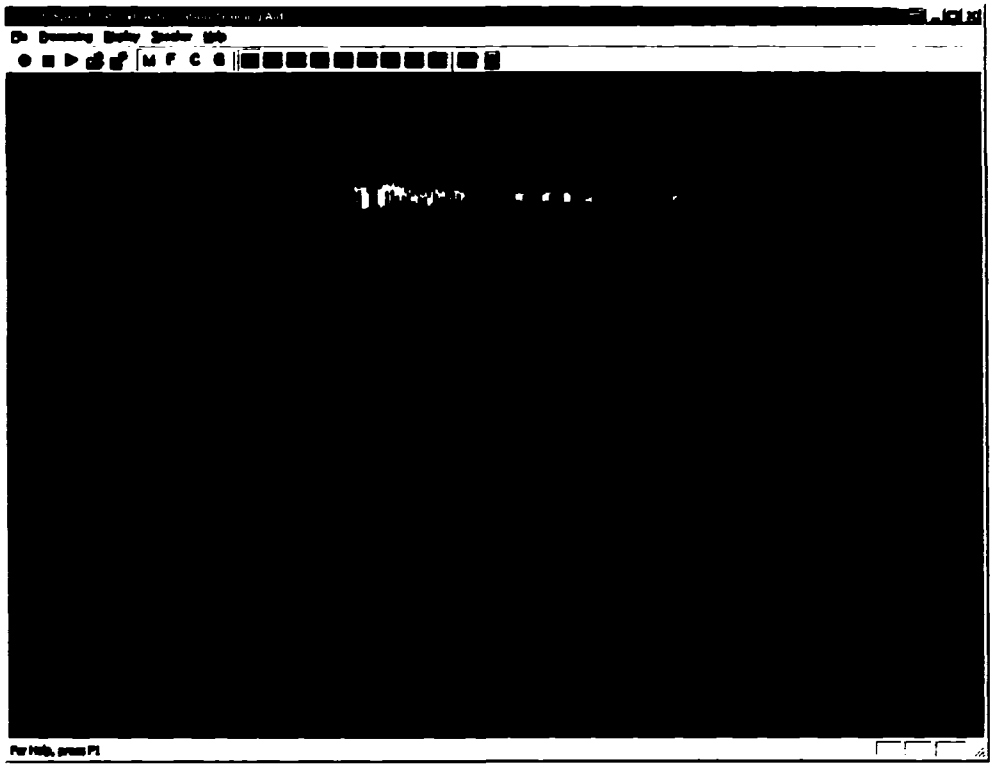

Fig. 23. Waveform display.

The display contains two panels. The top panel displays the entire waveform. The signal displayed in the bottom panel is a 60 ms segment of the signal from the top panel, selected to begin with the cursor shown in the top panel. In both panels, the horizontal axis indicates time and the vertical axis indicates amplitude. Under the non-real-time mode, users can select a time instant of interest by moving cursor in the top panel, and then the corresponding signal will be displayed in the bottom panel.

6.4.3 The short time spectrum display

The short time spectrum display gives a picture of the distribution of frequency and amplitude at a selected moment in time. It can be used to analyze speech signals in the frequency domain. An example is illustrated as Fig. 24.

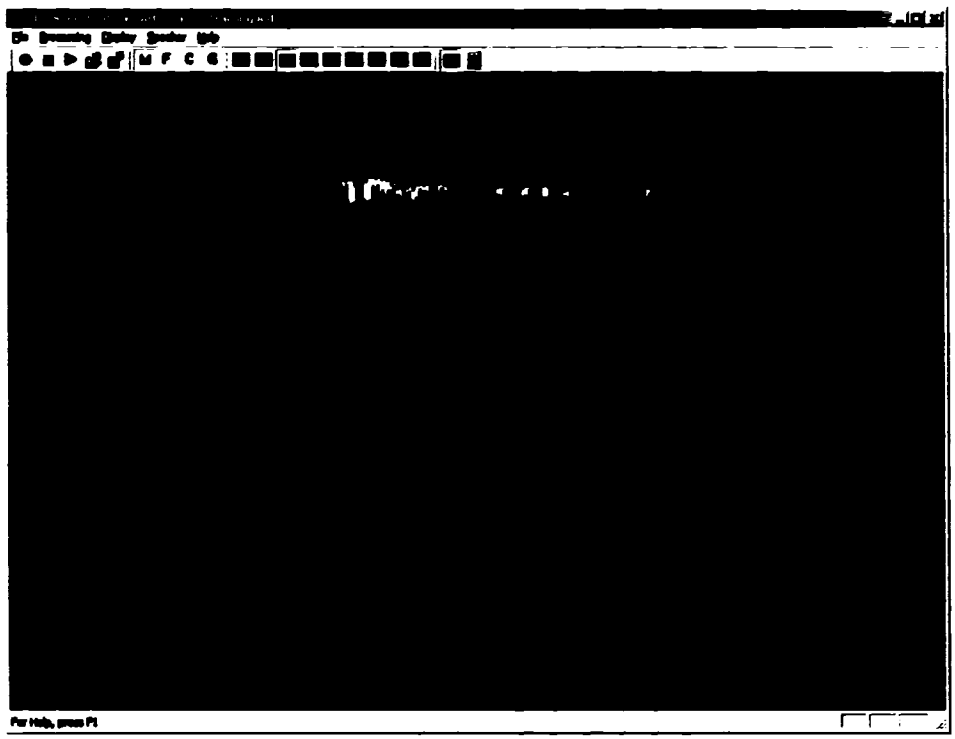

Fig. 24. Acoustic time signal (upper panel) and spectrum (lower panel).

The display also contains two panels; the top panel displays the overall time waveform. The frequency response is displayed in the bottom panel with a curve. Note that this panel does not have a time scale. Instead, the horizontal axis represents frequency, and the vertical axis amplitude. This display provides the ability for users to examine the frequency response at any time, again by moving the cursor shown in the top panel as mentioned previously. When a user selects a point from the top panel, the corresponding frequency response will be shown in the lower panel.

Two modes were designed for the frequency response display. One mode displays

only the current spectrum at the selected time moment; another mode displays three spectral curves simultaneously with red, yellow and blue colors. The red curve represents the current spectrum, the yellow and blue curve represent the spectra at the two previous time moments (that is the two previous frames). Users can learn about spectral changes by observing how these spectra change at different parts of a CVC.

6.4.4 The spectrogram display

A spectrogram is a display of the frequency content of a signal drawn so that the energy content in each frequency region and time is displayed on a colored scale. From such pictures, speech sounds can be observed from a spectral point of view. Such displays have been used by speech scientists for over 30 years. Fig. 25 and Fig. 26 gives two examples of the spectrogram for two different words by the same speaker.

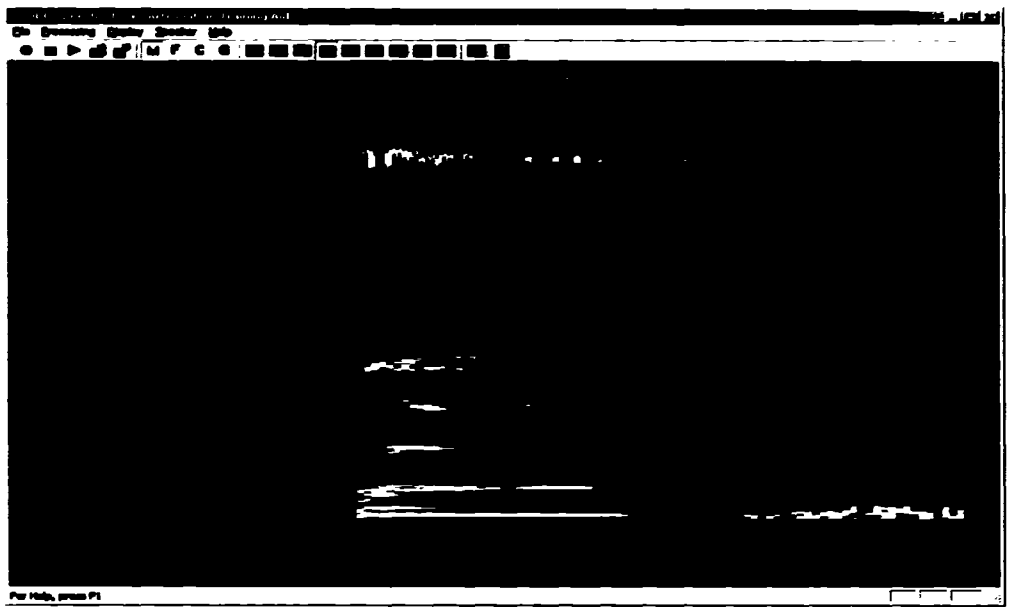

Fig. 25. Acoustic time signal (upper panel) and spectrogram (lower panel) to the word "bag" by a male speaker.

Fig. 26. Acoustic time signal (upper panel) and spectrogram (lower panel) to the word "boyd" by the same speaker as in Fig. 25.

The horizontal axis of the spectrogram is time, and the display shows how the signal develops and changes over time. The vertical axis of the spectrogram is frequency and it provides an analysis of the signal into different frequency regions. The third dimension, amplitude, is represented by colors. In this dissertation, the higher values are represented with warm colors (like red) while the lower values are represented with cold colors (like blue). Consider the spectrogram to be a number of spectrums in a row, looked upon "from above", and where the highs in the spectra are represented with a red color in the spectrogram.

In addition to the spectrograms showed above, a "smoothed" spectrogram (Fig. 27) also was developed. In this spectrogram, the spectrum at each time moment was obtained by recomputing the spectrum from DCTC features. When computing a DCTC feature vector with N elements, only the first N coefficients were kept; the remaining

terms were discarded. The spectrums recovered from the first N coefficients are equivalent to applying a low pass filter to the original spectrum.

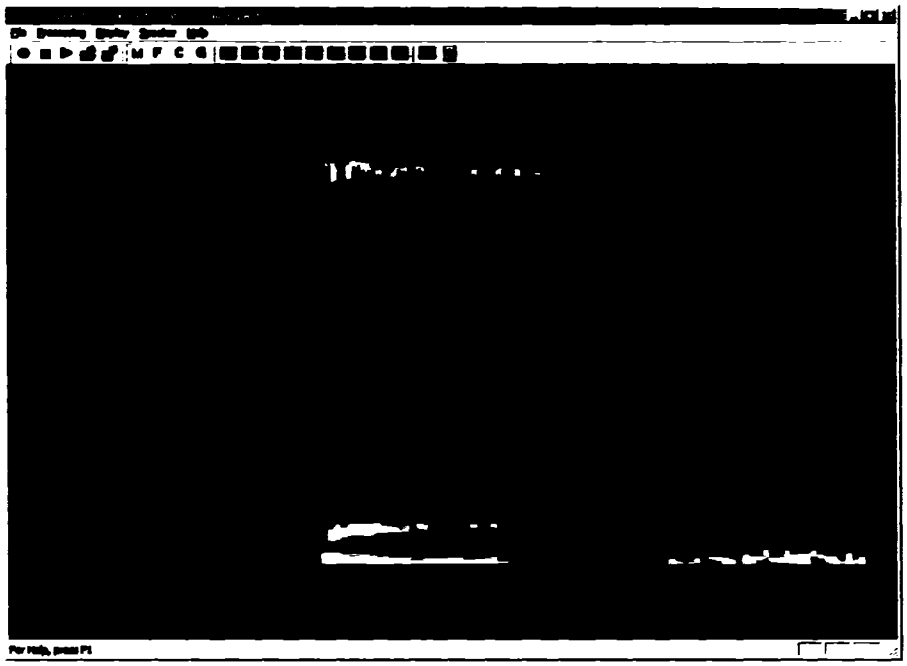

Fig. 27. Acoustic time signal (upper panel) and smoothed spectrogram (lower panel).

6.4.5 The feature bargraph display

In this display, the values of each feature components at a time moment are represented by the height of bars. Here the values of feature components have been normalized to the range from -1 to 1. An example is illustrated as Fig. 28.

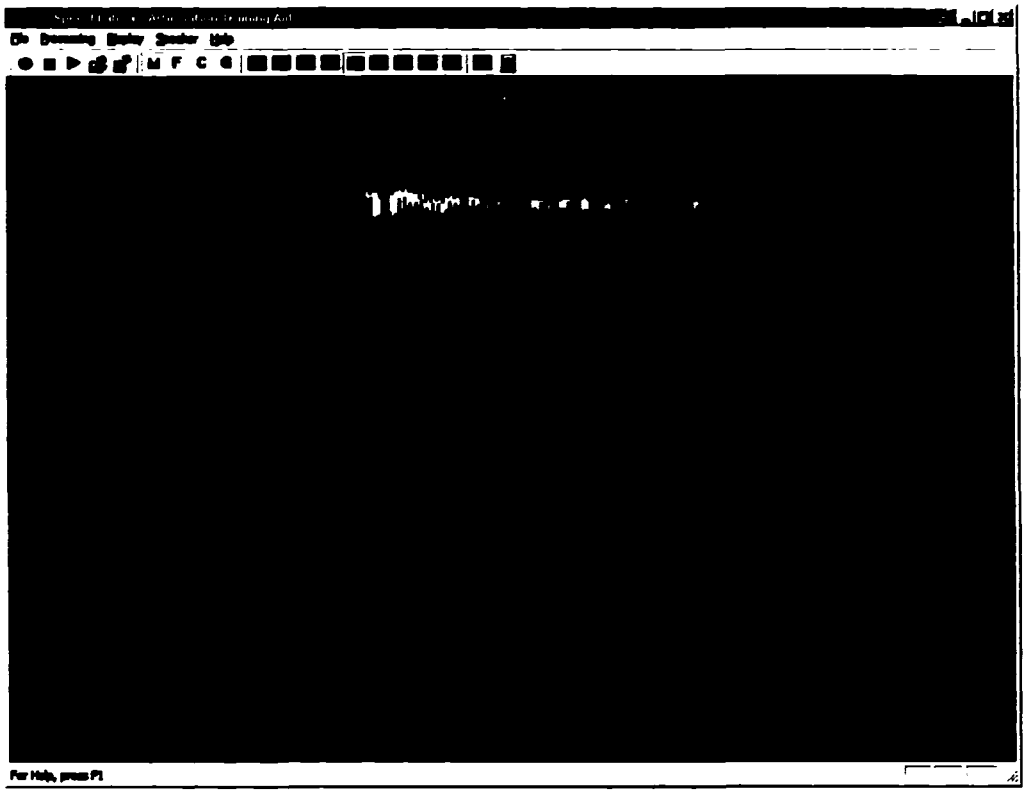

Fig. 28. Acoustic time signal (upper panel) and DCSC features (low panel).

This display, which is similar to other displays, has two panels. The top panel contains the speech waveform. The user can select a time moment of interest by moving the cursor. The values of features are displayed in the lower panel. The horizontal axis is the feature index and the vertical axis is amplitude. There are three kinds of bars with red, yellow and green color to represent the feature properties. The red bars indicate DCTC coefficients; the yellow bars indicate first order DCS coefficients and the green bars indicate the second order of DCS coefficients.

6.4.6 The bargrph display

Bargraph display gives the phonetic recognition results for a given speech feature vector at a particular time moment. The recognizer is a three layer neural network with 20 outputs. Fig. 29 shows the neural network result for a CVC word at a vowel time instant.

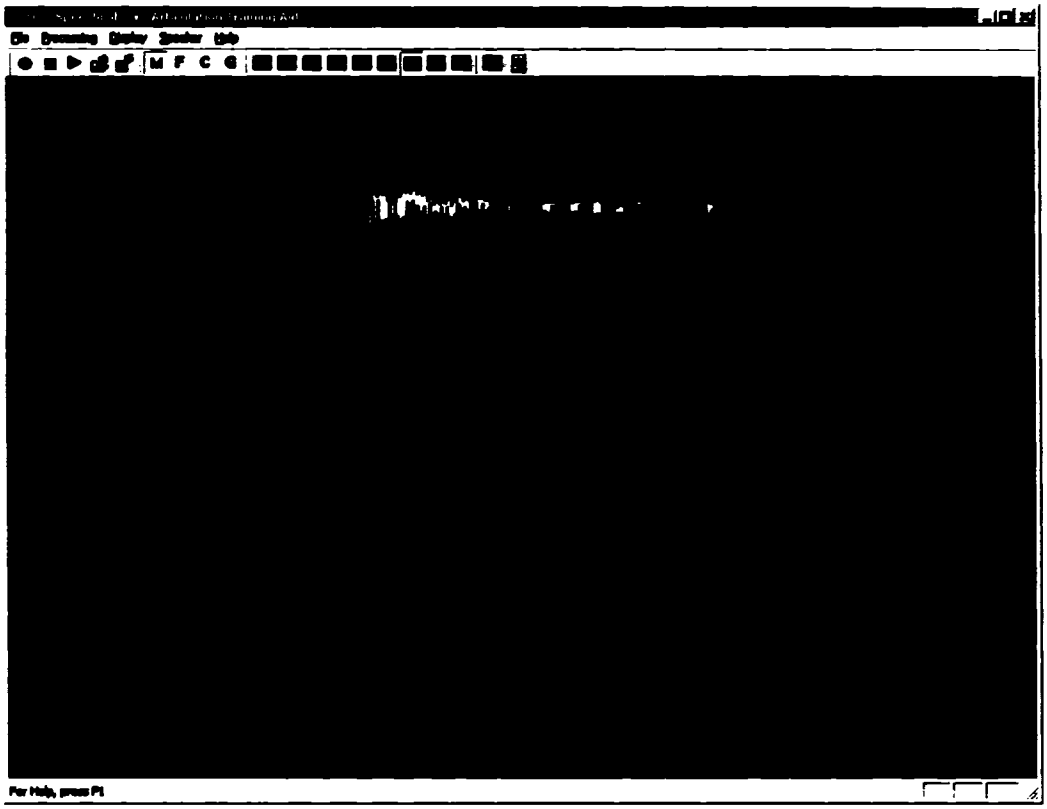

Fig. 29. Acoustic time signal (upper panel) and phonetic recognition results for a specified block (low panel).

As for other displays, users can select a time moment in the top panel, and the neural network recognition result for that time moment is shown is the lower panel. Each column indicates a phoneme category and categories are also represented with different colors. In Fig. 29, the speech segment at the selected time moment (represented with the red line) is recognized as "ae;" however, it is also partially recognized as a similar sound "aa".

6.4.6 The word recognition display

The word recognition display gives recognition results for a given CVC word. The display is illustrated as Fig. 30

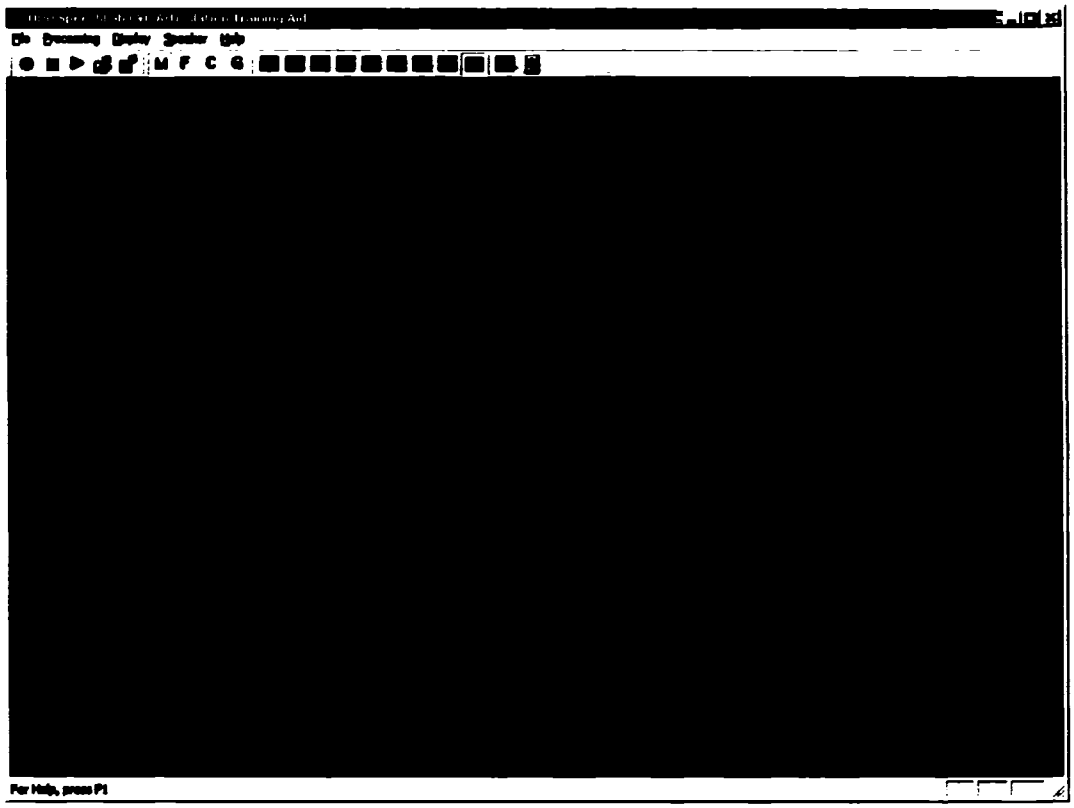

Fig. 30. Word recognition display.

The display contains 13 red blocks and each block indicates a CVC word. Users input their voice through a microphone, and the recognition result will be displayed by highlighting the corresponding block.

6.5 Evaluation

The system has been developed and it was evaluated with recorded CVC sounds and by the people in the laboratory. The system works well for the word level recognition. However, this system has not been tested for the people with hearing impairments. Much more extensive testing is needed in future work.

CHAPTER VII

CONCLUSION AND SUGGESTIONS FOR FUTURE WORK

In this dissertation, a system of computer-based speech training aids for the speech and hearing impaired was developed. The system is intended to help children to improve their pronunciation of isolated CVC words at both the phone level and word level. The system provides real time response to a users voice input as well as well as abundant and easy-to-interpret information. Thus users can learn how to improve their incorrect pronunciations.

Speech recognition accuracy plays an important role in speech training. The speech training application requires satisfactory speech recognition accuracy at both the phonetic level and word level. Two hybrid HMM/NN methods for speech recognition were investigated. The first method uses neural networks as posterior probability estimators for HMMs. In the second method, the neural networks transform the original speech feature to normalized speech features. Compared with original speech features, the normalized features have less correlation and thus better satisfy one assumption for HMMs.

7.1 Contributions

In chapter 4, a hybrid HMM/NN method was presented. This method uses neural network outputs as posterior probability estimators for HMMs. Some experiments were performed with different features and variable settings for each feature. From the experimental results, the following conclusions can be drawn:

1. By adjusting the variable of "block length" and number of DCSs for computing DCSC features, a better recognition rate can be obtained with DCSC features than with MFCC features.
2. Comparing the results of the hybrid method to those of standard HMM/GMM methods, the hybrid method with DCSC features outperforms HMM/GMM methods with both MFCC features and DCSC features.
3. Similar results can be obtained for two neural network architectures --- MLP and TDNN. However, MLP has a simpler architecture than that of TDNN.

In chapter 5, a new hybrid of HMM/NN method was presented. The neural networks transform the original speech feature to normalized speech features and the HMMs are trained with normalized features. In theory, the normalized speech features have less correlation than that of original speech features. This theory was illustrated by experimental results presented in chapter 5. Additionally, these normalized features are more compact than the original features. The experimental results at the word level showed that this method outperforms the standard HMM/GMM method and the hybrid method presented in chapter 4.

In chapter 6, a PC based CVC display system was described. The system is able to visually present indicators of pronunciation "correctness" at the phone and word level, in response to short words produced by a speaker. The recognizer in the system was created by applying the recognition method described in chapter 4. In chapter 6, more emphasis is given to a description of the graphical outputs. A flow-mode display and other displays are discussed, all of which can be used for speech articulation training.

7.2 Future Work

There are several suggestions for further work as follows:

1. The training data and test data for this dissertation were recorded in a clean environment. In practice, the users may use this system in noisy environments. An important and interesting research field is a study of adaptation to different environment noises.
2. The performance of the algorithms presented in this dissertation still could be improved. More studies are needed to develop more accurate and robust speech recognition algorithms.
3. It would help to improve the system's displays to provide more useful and clearer information so that users could learn to correct their incorrect pronunciation more efficiently.
4. Because of time and other limitations, the system was not evaluated by actual users. Formal evaluation with people who are hearing impaired should be conducted to test and improve the system.
5. The current system only works for a small vocabulary of CVC words. The vocabulary size should be increased and the system should be made to function for continuous speech.
6. The system has been placed on the web. Anyone may freely download the program and install it on his/her computer. To better spread the usage of this system, it would help to have a web-based system. Then people could access and use the system directly through the internet.

LITERATURE CITED

[1] M. Ostendorf, E. Shriberg and A. Stolck, "Human language technology: opportunities and challenges," *Proc. ICASSP*, vol. 1, pp. 949-952, 2005.

[2] D. Deroo, "A short introduction to speech recognition," http://www.babeltech.com.

[3] N. Morgan and H. Bourlard, "An introduction to hybrid HMM/Connectionist Continuous Speech Recognition", *IEEE Signal Processing Magazine,* pp. 25-42, May, 1995.

[4] L. Deng, D. Yu and A. Acero., "A quantitative model for formant dynamics and contextually assimilated reduction in fluent speech," *Proc. ICSLP*, vol. 1, 981-984, Jeju Island, Korea, 2004.

[5] W. Zhu and O'Shaughnessy "Log-Energy Dynamic Range Normalization for Robust Speech Recognition", *Proc. ICASSP*, vol. 1, pp. 245-248, 2005.

[6] M. Bacchiani, B. Roark and M. Saraclar, "Language model adaptation with MAP estimation and the perceptron algorithm," *Proceedings of the HLTNAACL*, vol. 1, pp. 21-24. Boston, MA, May 2004.

[7] S. Anderson and D. Kewley-Port "Evaluation of Speech Recognizers for Speech Training Applications," *IEEE Trans. on Speech and Audio Processing*, vol. 3, No. 4, pp 229-241, 1995.

[8] KJ. Munro and ME. Lutman. "The influence of visual feedback on closed-set word test performance over time," *Int J Audiol,* vol. 44, pp.701-705, Dec 2005.

[9] J. L. Braeges and R. A. Houde, "Use of speech training aids," *Deafness and Communication, Assessment and Training*, (edited in Sims, D., Walter, G., & Whitehead, R.) pp. 222-244, 1981.

[10] D. Kewley-Port, C.S. Watson, et al., "The Indiana Speech Training Aid (ISTRA) II: Training curriculum and selected case studies," *Clinical Linguistics and Phonetics*, vol.5, pp. 13-38, 1991.

[11] L. Rabiner and B.H.Juang, *Fundamentals of Speech Recognition*, Prentice Hall, Englewood Cliffs, 1993.

[12] L. Rabiner and R. Schafer, *Digital Processing of Speech signal*, Prentice-Hall, Englewood Cliffs, 1978.

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

[14] S. Zahorian and A. Jagharghi, "Spectral-shape Features versus Formants as Acoustic Correlates for Vowels", *J. Acoust. Soc. Amer.*, vol. 94, pp 1966-1982, 1992.

[15] R. J. Trybus, "National data on rated speech intelligibility of hearing-impaired children," *Speech assessment and speech improvement for the hearing impaired*, edited in J. Subtelny (A. G. Bell Association for the Deaf, Washington, D. C.) pp. 67-71, 1980.

[16] D. Ling, *Speech and the hearing-impaired child: Theory and practice*, (The Alexander Graham Bell Association for the Deaf, Washington, D.C. ISBN 0-88200-074-8), 1976.

[17] P. A. Dagenais, "Speech Training with Glossometry and Palatometry for Profoundly Hearing-Impaired Children," *Volta Review*, vol. 94, pp. 261-82, 1992.

[18] H. Levitt, "Technology and Speech Training: An Affair to Remember", *Volta Review,* vol. 5, pp. 1-5, 1991.

[19] R. E. Stark, "Speech of the hearing-impaired child," *Hearing and Hearing Impairment*, (edited in L. J. Bradford and W. G. Hardy), pp. 229-248, 1979.

[20] D. D. Johnson, "Communication characteristics of a young deaf adult population: Techniques for evaluating their communication skills," *Amer. Annals of the Deaf*, Appendix B, August, 1976.

[21] B. Schimmer, *Language and Literacy Development in Children Who are Deaf,* Merrill Publishing Co., New York,1994.

[22] J. D. Subtelny, "Integrated speech and language instruction for the hearing-impaired adolescent," *Speech and language: Advances in basic research and practice,* edited by N. J. Lass (Academic Press, 1983, NY), Vol 9, pp. 43-102, 1983.

[23] Tye_Murray, N., "*Foundations of Aural Rehabilitation*", Singular Publishing Group, San Diego, Ca., 1998.

[24] Ling, D., *"Speech and the hearing-impaired child: Theory and practice*", The Alexander Graham Bell Association for the Deaf, Washington, D.C. ISBN 0-88200-074-8), 1976.

[25] R. S. Nickerson, D. N. Kalikow and K. N. Stevens, "Computer-aided speech training for the deaf," *Journal of Speech and Hearing Disorders* vol. 61, pp. 120-132, 1976.

[26] D. Kewley-Port and C. S. Watson, "Computer assisted speech training: Practical considerations," edited by A. Syrdal, R. Bennett, and S. Greenspan, Applied Speech Technology, Boca Raton, FL: CRC Press, pp. 565-582, 1994.

[27] D.J. Povel and N. Arends, "The Visual Speech Apparatus: Theoretical and practical aspects," *Speech Communication*, vol. 10, pp. 59-80, 1991.

[28] IBM Speech viewer III, http://my.execpc.com/~labres/ibm.html.

[29] Video Voice, http://www.videovoice.com/vv_choos.htm.

[30] TC-Helicon, Voice Prism, http://www.tc-helicon.com.

[31] H.T. Bunnell, D. Yarrington and J.B. Polikoff, "STAR: Articulation Training for Young Children." *Proceedings of the Sixth International Conference on Spoken Language Processing*, Beijing, China, vol. 4, pp. 85-88, 2000.

[32] A-M. Öster, H. David, A. Protopapas and A. Hatzis, "*Presentation of a new EU project for speech therapy: OLP (Ortho-Logo-Paedia)*" in: Fonetik, vol. 44, pp. 45-48, 2002.

[33] S. Zahorian, M. Zimmer and Meng F., "Vowel Classification for computer-based visual feedback for speech training for the hearing impaired", *ICSLP*, vol. 1, pp.973-976, 2002.

[34] A. Zimmer, B. Dai and S. Zahorian, "Personal Computer Software Vowel Training Aid for the Hearing Impaired", *proc. ICASSP*, vol. 6, pp. 3625-3628, 1998.

[35] D.J. Povel and N. Arends, "The Visual Speech Apparatus: Theoretical and practical aspects," *Speech Communication*, vol. 10, pp. 59-80, 1991.

[36] S. Anderson and D. Kewley-Port, "Evaluation of Speech Recognizers for Speech Training Applications," *IEEE Trans. on Speech and Audio Processing*, vol. 3, no. 4, pp 229-241, 1991.

[37] J.J. Mahshie, "Balloons, Penguins, and Visual Displays," *Perspectives in Education and Deafness*, vol.16, no. 4, 1998.

[38] L. Cleuren, "Speech Technology in Speech Therapy? -- State of the Art and Onset to the Development of a Therapeutic Tool to Treat Reading Difficulties in the First Grade of Elementary School". masters thesis, K.U.Leuven, ESAT., MAI project report, 2003.

[39] S. Evangelos, N. Fakotakis and G. Kokkinakis, "Fast endpoint detection algorithm for isolated word recognition in office environment", *proc. ICASSP,* vol. 1, pp. 733-736, 1991.

[40] M. Devarajan, F. Meng, P. Hix, and S. Zahorian, "HMM-Neural Network monophone models for computer-based articulation training for the hearing impaired," *proc. ICASSP* vol. 2, pp. 369-372, 2003.

[41] L. Baum, "An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov Process", *Inequalities*, vol. 3, pp. 1-8, 1972.

[42] F. Jelinek, *statistical methods for speech recognition*, The MIT press, Cambridge, Massachusetts, 1999.

[43] R. Duda and P. Hart, *Pattern Classification*, Second Edition, Wiley, 2001.

[44] J. Andrew, "Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, pp. 260 -269, 1967.

[45] Young et al., "*Hidden Markov Model Toolkit v3.1 reference manual*", Technical report, Speech group, Cambridge University Engineering Department, December 2001.

[46] R.P. Lippmann, "Review of neural networks for speech recognition," *Neural Comput,* vol. 1, pp. 1-38, 1989.

[47] Y. Bengio, "Radial basis functions for speech recognition, in Speech Recognition and Understanding: Recent Advances, Trends and Applications", NATO *Advanced Study Institute Series F: Computer and Systems Sciences*, pp. 293-298, 1990.

[48] J. Hosom, R. Cole, et al. "Training Neural Networks for Speech Recognition Center for Spoken Language Understanding", *Oregon Graduate Institute of Science and Technology*, February, 1999.

[49] D. E. Rumelhart, G. E. Hinton and R. J. Williams. "Learning internal representations by back-propagating errors." *Nature,* vol. 323 pp. 533-536, 1986.

[50] R.P. Lippmann and B. Gold, "Neural classifiers useful for speech recognition," *IEEE Proceedings of First International Conference on Neural Networks*, vol. IV, pp. 417-422, San Diego, CA, 1987.

[51] A. Waibel, H. Sawai and K. Shikano, "Modularity and scaling in large phonemic neural networks," *IEEE Trans. Acoust. Speech Signal Process.* vol. 37, pp. 1888-1898, 1989.

[52] P. Haffner, A. Waibel, and K. Shikano, "Fast back-propagation learning methods for large phonemic neural networks," *Proceedings of Eurospeech*, 1989.

[53] D.E. Rumelhart, G.E. Hinton and G.E. Williams, "Learning internal representations by error propagation," *Parallel Distributed Processing*, vol. 1. pp. 318-362 (Chapter 8), 1986.

[54] A. Waibel, "Modular construction of time-delay neural networks for speech recognition," *Neural Comput,* vol. 1, pp.39-46, 1989.

[55] A. Waibel, T. Hanazawa, et al. "Phoneme recognition using time-delay neural networks," *IEEE Trans. Acoust. Speech Signal Process,* vol. 37, pp.328-339, 1989.

[56] M. Franzini, M. Witbrock, and K.F. Lee "Speaker-Independent Recognition of Connected Utterances using Recurrent and Non-Recurrent Neural Networks," *In Proc. International Joint Conference on Neural Networks*, vol. 2, pp. 1-16, 1989.

[57] J.L. Elman, "Finding structure in time," *Cognitive Sci.* vol. 14, pp. 179-211, 1990.

[58] G. Zavaliagkos, Y. Zhao, et al. "A hybrid segmental neural net/hidden Markov model system for continuous speech recognition," *IEEE Trans. Speech Audio Process.* vol. 2, pp 151-160, 1994.

[59] D. Yu, T. Huang and D.W. Chen, "A multi-stage NN/HMM hybrid method for high performance speech recognition," *proc. ICSLP*, vol.2, pp. 1503-1506, 1994.

[60] J.S. Bridle, "Alphabets: a recurrent & neural network architecture with a hidden Markov model interpretation," *Speech Commun,* vol. 9, pp. 83-92, 1990.

[61] S.K. Riis, and A. Krogh, "Hidden neural networks: a framework for HMM/NN hybrids", *proc. ICASSP*, vol. 4, pp. 3233-3236, 1997.

[62] G. Rigoll, "Hybrid Speech Recognition Systems: A Real Alternative to Traditional Approaches". *In Survey Lecture, Int. Workshop "Speech and Computer"*, pages 33-42, 1998.

[63] G. Rigoll, "Maximum mutual information neural networks for hybrid connectionist-HMM speech recognition systems," *IEEE Trans. Speech Audio Process.* vol. 2, no. 1, pp. 175-1184, 1994.

[64] C.S. Jang and C.K. Un, "A new parameter smoothing method in the hybrid TDNN/HMM architecture for speech recognition", *Speech Communication,* vol. 19, no. 4, pp. 317-324, 1996.

[65] H. Bourlard, and N. Morgan, "Continuous speech recognition by connectionist statistical methods", *IEEE Trans. Neural Networks*, vol. 4, no. 6, pp. 893-909, 1993.

[66] E. Trentin and M. Gori, "A survey of hybrid ANN/HMM models for automatic speech recognition". *Neurocomputing,* vol. 37, no. 1-4, pp. 91-126, 2001.

[67] M.D. Richard and R.P. Lippmann, "Neural network classifiers estimate Bayesian a posterior probabilities," *Neural Computation*, vol. 3. no.4, pp. 461-483, 1991.

[68] E. Singer and R.P. Lippmann, "A speech recognizer using radial basis function neural networks in an HMM framework," *proc. ICASSP*, vol. 1, pp. 629-632, 1992.

[69] T. Robinson, "A real-time recurrent error propagation network word recognition system," *proc. ICASSP*, vol. 1, pp. 617-620, 1992.

[70] Y. Yan, M. Fanty and R. Cole, "Speech recognition using neural networks with forward-backward probability generated targets," *proc. ICASSP*, vol. 4, pp. 3241-3244, 1997.

[71] K. Kirchhoff and J. Bilmes, "Dynamic Classifier Combination in Hybrid Speech Recognition Systems using Utterance-Level Confidence Values," *proc. ICASSP*, vol. 1, pp. 693-696, 1999.

[72] J. Tebelskis, "Speech recognition using neural networks," master thesis, CMU 1995.

[73] A. Papoulis *Probability, Random Variables and Stochastic Processes (3rd edition)*, McGraw Hill. February, 1991.

[74] J. G. Proakis and D.G. Manolakis, *Digital Signal Processing, Digital Signal Processing: Principles, Algorithms, and Applications (3rd edition)*, Prentice Hall, February, 1995.

[75] S. Zahorian, and K. Kasi, "Yet another Algorithm for Pitch Tracking", vol.1, pp.361-364, *proc. ICASSP*, 2002.

[76] B. J. Lindbloom, "A Continuity Study of the CIE L* Function", http://www.brucelindbloom.com, 2004.

[77] P. Colantoni et al., "Color Space Transformations", Documentation to the program ColorSpace, http://www.couleur.org, 2004.

CURRICULUM VITA
For

Fansheng Meng

EDUCATION

M. S. Electrical Engineering, University of Electronic Science & Technology of China, China

B. S. Electrical Engineering, University of Electronic Science & Technology of China, China

PAPERS PUBLISHED

- Fansheng Meng, Stephen Zahorian, "Hybrid HMM/NN Methods with DCSC Speech Features for Isolated Word Recognition", submitted to Interspeech 2006
- Penny Hix, Stephen Zahorian, Fansheng Meng, "Novel feature extraction for noise robust ASR using the AURORA2 database", accepted by *ICASSP* 2006.
- Fansheng Meng, Eugen Rodel, and Stephen Zahorian, "A consonant-vowel-consonant display as a computer-based articulation training aid for the hearing impaired", *The Journal of the Acoustical Society of America*, 2005 Volume 117, Issue 4, p. 2621
- Stephen A. Zahorian, Mame Sall, Fansheng Meng, and Wei Wang, "Generalization of Support Vector Machines versus Neural Networks for Pattern Classification," *Intelligent Engineering Systems through Artificial Neural Networks* , Volume 14, pp 639-644, Nov 7-10, 2004,
- Devarajan, M., Meng, F., Hix, P., and Zahorian, *S. A.*, "HMM-Neural Network monophone models for computer-based articulation training for the hearing impaired", *ICASSP* 2003
- Zahorian S., Zimmer M., Meng F., "Vowel Classification for computer-based visual feedback for speech training for the hearing impaired", *International Conference on Spoken Language Processing, 2002.*

WORK EXPERIENCE

Yantai Dongfang Electronics Information Industry Co., Ltd. Yantai, Shandong, China. 1998-2000