A COMPUTER-BASED ARTICULATION TRAINING AID FOR SHORT WORDS (CATA)

by

Mukund Devarajan
B.E. May 1998, Madras University, India

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

MASTER OF SCIENCE

ELECTRICAL ENGINEERING

OLD DOMINION UNIVERSITY
December 2003

Approved by:

Stephen A. Zahorian (Director)

Vijayan K. Asari (Member)

Min Song (Member)

ABSTRACT

A COMPUTER-BASED ARTICULATION TRAINING AID FOR SHORT WORDS (CATA)

Mukund Devarajan
Old Dominion University, 2003
Director: Dr. Stephen A. Zahorian

Several improvements in the vowel articulation training aid (VATA) are described, as well as the efforts to extend the visual feedback system to operate with short words in the form of consonant, vowel and consonant (CVC). The extended version of the visual feedback system is referred to as CATA (Computer-based Articulation Training Aid); the vowel version of the aid (VATA) only operates with ten American English monopthong vowels. Improvements in VATA include the use of a neural network (NN) recognizer method to prune a large database of vowel recordings to eliminate noisy and/or mispronounced tokens. The spectral jitter problem, previously present in the VATA, has also been corrected. Initial steps in the development of CATA involved database preparation. The training methodologies and the step-by-step procedure for using Hidden Markov Modeling (HMM) for recognizing and segmenting a CVC database are described. The signal processing and recognition steps involved in building a real-time display system to provide visual feedback about the quality of pronunciation of the CVCs are described in detail. An attempt at using a time-delay neural network (TDNN) classifier for distinguishing phonemes present in the CVCs is described. Experiments conducted to improve the VATA and the initial results obtained with the CVC display system are reported.

To my lovely Mom and wonderful Dad!

Words are not enough to describe your love and support

ACKNOWLEDGMENTS

I would like to thank Dr. Stephen A. Zahorian for providing an opportunity to work in the Old Dominion University’s speech communication lab. It would have been next to impossible to complete this thesis if not for him. I owe the speech signal processing knowledge that I gained over the period of doing research in this lab to his constant guidance.

I would also like to express my thanks to Mr. Fansheng Meng for his useful suggestions and assistance and to Mr. Daniel Vasconcelos for helping with the speech database cleanup during the course of this thesis research.

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

Page

Chapter

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

OVERVIEW OF COMPUTER-BASED VISUAL TRAINING AIDS

Introduction

Teaching oral speech to hearing-impaired people has been the subject of fundamental importance to speech signal processing researchers and speech therapists alike for the past seventy years. However, methods for teaching have not yet been perfected. Hearing-impaired people have problems with their oral speech thus making it difficult to comprehend their speech. The main reason for this is because of their inability to hear what they say. As a result of this, they are not able to "perceive" what they speak and make corrections as needed, as can a normal individual with no hearing disorder. Hence, there is a need for external feedback to act as a substitute for hearing and complete the sequence of "sense", "perceive" and "act". Visual training aids have been proposed to act as these external elements and have been in existence for decades. Nevertheless, the success rate achieved for improving the speaking quality for hearing-impaired children is far less than phenomenal. This may be due to the fact that these visual aids failed to comply with one or more of the basic rules proposed by Hudgins [1]:

"(1) The visual pattern must be simple and clear-cut so that the deaf child will have no difficulty in understanding it; (2) the apparatus which presents these visual patterns must be easy to operate and adaptable to the classroom; (3) the apparatus must present the visual patterns while the child is speaking".

Visual aids provide either "process-oriented" information or the "product-oriented" one [2]. Process-oriented information-based visual training aids provide information about the process of generating speech by providing displays of method of articulation, vocal tract etc. which serve as a reference for the hearing-impaired. The product-oriented ones provide speech-product models in the acoustic domain; the hearing-impaired try to match the models and the method of learning is complete when there are no discrepancies between the intended speech and that produced. Product-based visual training aids are found to be more useful compared to the process-oriented one as they are found to be a natural way of learning and easy to comprehend, especially for hearing-impaired, particularly children.

This thesis uses journal model of IEEE Transactions on speech and audio processing for tables, figures and references

History of Visual Training Aids

The first computer-based visual training aid was developed over 30 years ago [3][4]. Results indicated that, though some aspects of speech related to supra-segmental features improved, there were no gains as far as conversational speech is concerned. This provided the motivation for the future visual training aids to emphasize articulatory-phonetic training as opposed to supra-segmental speech features. Since then quite a few visual speech training aids aimed at correcting articulatory behavior have been developed and documented [5] – [7]. The following is a brief overview of three visual speech-training aids that were developed in the late 1980's and the early 1990's:

Visual Speech Apparatus (VSA)

VSA is a speaker-dependent visual speech apparatus that displays all aspects of speech relevant for speech training that can be learned naturally. It is easily understandable and is designed to augment the role of the speech therapist [2]. It consists of a Ariel DSP-16 and the Commodore Amiga 2000 computer with a PC/XT board based on 8088 processor emulating an IBM-PC computer. Microphone and Laryngograph serve as inputs of an audio-interface to the DSP-16 processor. The DSP-16 processor performs the sampling and computations on the incoming signals to produce the outputs which are displayed on the screen.

VSA has two modes of operation viz. LESSON-mode and HOMEWORK-mode. The LESSON-mode is where the pupil and the therapist work together; therapist selects the exercises and the order of performance during a lesson. In the HOMEWORK-mode, the pupil works alone and performs the exercises assigned by the therapist.

VSA uses single-dimension displays. There are two types of displays: displays for calibration and another for the exercises. Examples of the displays used for calibration include the oscillogram and the laryngogram. The one designed for exercises and training include the ones for timing, loudness and pitch which are displayed with time as location on the X-axis, loudness as the size and brightness of an object, pitch as the vertical location of the object on the screen and voicing as the color. VSA also includes a variant of the Vowel Corrector [7][8]. This vowel display gives an intuitive idea about the relation between the vowels and the vowel tract configurations. The individual exercises like those intended for training voicing, loudness, pitch etc are designed to impart subskills that are selected serially and designed towards a teaching target. The various subskills can be then integrated to establish the sound-meaning relation by the speech pathologist.

Indiana Speech Training Aid (ISTRA)

ISTRA is a commercially available speaker-dependent visual speech training system developed to serve as a valuable tool to a speech therapist [9]. It is a product of several years of research and, hence, is moderately advanced. It is implemented on a PC-type microcomputer equipped with a speaker-dependent recognition board.

Visual feedback is given by means of template matching that is derived from an estimate of the similarity between the speaker-produced sound and the stored template of the same utterance. These templates are continually updated with the best recent effort so that the speech intelligibility and the final goal of improvement of speech pronunciation are achieved.

As in the case of any typical visual aid, speech drills are presented in the form of video games with a few of them having the option of being used at different training levels. The speech pathologist is provided with a speech-training curriculum to administer that contains a series of speech training drills which can be used either for diagnostic or treatment methods. Records of all training for a given speaker can be stored on a floppy disk and can be viewed on the screen or printed by the speech pathologist.

IBM Speech Viewer III

The IBM Speech Viewer III is a commercially available speaker-dependent visual speech training aid that consists of a collection of software programs which run on a typical PC running on Windows (3.X, 95, 98, ME, 2000, XP). It is intended to provide a solid tool to the speech therapist. It requires an IBM-supported industry soundcard (typically soundblaster 16) and uses a (standard) microphone as the input. It is similar to any windows-based program and, hence, easy to operate.

Feedback is provided immediately by a number of visual displays intended to improve areas such as voicing, pitch, loudness, timing patterns and sustained vowel and consonant production. Several game displays have been included so that children do not loose interest easily. Also included are a number of calibration displays such as waveform and sound spectra. Loudness, for example, is taught by means of a display called “Balloons” where a boy blows up a red balloon in the screen with the size of the balloon proportional to the loudness of the speaker. Another exercise is for pitch control; the speaker must vary his or her pitch to hit a target without colliding with an obstacle. In order to recognize the phonemes (vowels and consonants), the system must first be taught so that it has a model of the phoneme. The system is included with an easy to use utility program where a set of templates can be generated for each phoneme either using the hearing-impaired child’s own productions or those of a hearing-child or those of a speech therapist. This need for custom training is one of the known drawbacks in the system because of the difficulty involved in producing adequate data for the training process. Overall, Speech Viewer III does a reasonably good job of providing accurate feedback about timing, voicing, pitch and loudness.

General Overview of Vowel Articulation Training Aid (VATA)

VATA is a Win32 platform-based speaker-independent isolated-vowel recognizer designed to function as a speech therapy tool to compensate for the insufficient or missing auditory feedback for hearing impaired persons [10]. It is implemented on a typical PC running windows (95,98,Me, NT, 2000,XP) with a soundcard and a microphone. It gives a visual display about the quality of pronunciation of ten American English monopthong vowels /ah/, /ee/, /ue/, /ae/, /ur/, /ih/, /eh/, /aw/, /uh/ and /oo/. Note the vowel notation mentioned in the previous sentence is that used in the actual display, which was selected based on phonetic qualities. The ARPABET notations for these vowels are /aa/, /iy/, /uw/, /ae/, /er/, /ih/, /eh/, /ao/, /ah/ and /uh/.

System Description

VATA has two main displays: a bargraph display and an ellipse display. The bargraph display is similar to a histogram and gives feedback about the goodness-of-fit to ten vowel categories with the size of the bar corresponding to the goodness of the pronunciation of the vowel. The ellipse display provides continuous visual feedback about the utterances. The ten monopthong vowels are realized as elliptical regions with their locations and sizes roughly resembling the F1-F2 formant space [11]. A basketball icon is placed at that region enclosed by the vowel category correctly recognized by the utterance. However, incorrect pronunciations result in the ball wandering or coming to stop at the area not enclosed by the ellipse. By observing the movement of the ball continuously, the user learns how to control his/her pronunciation to get the ball in the correct ellipse (vowel). Three game displays (to attract children) namely pacman, tetris and chicken crossing road have been developed.

Signal Processing

A block diagram of VATA signal processing is shown below (fig. 1). Speech segments are collected via the soundcard and are broken down into frames and prefiltered by a high-frequency preemphasis filter centered at 3.2kHz. Spectral calculation is then done by calculating the Fast Fourier Transform (typically 512 points) and then non-linearly scaled by means of the logarithmic function. This log-scaled spectrum is then averaged over several frames (5-10 typically) and the Discrete Cosine Transform Coefficients (DCTCs) are calculated. These DCTCs (typically 12) are block encoded by a Discrete Cosine Series (DCS) expansion to encode the feature variations over time [12]. Since (at least ideally) there are no variations over time for steady-state vowels, the number of DCSs is generally fixed at 1. The DCTCs are then normalized (zero mean and standard deviation of ±0.2) and input to the neural network (NN) classifier. The NN is a feed-forward multi-layer perceptron and has one input node per feature (totally 12 input nodes), typically 25 nodes in the hidden layer and 10 output nodes corresponding to the 10 vowel categories and are typically trained using error backpropagation for 250,000 iterations. More details on the data management and real-time operation characteristics can be found in Zimmer's [13] thesis.

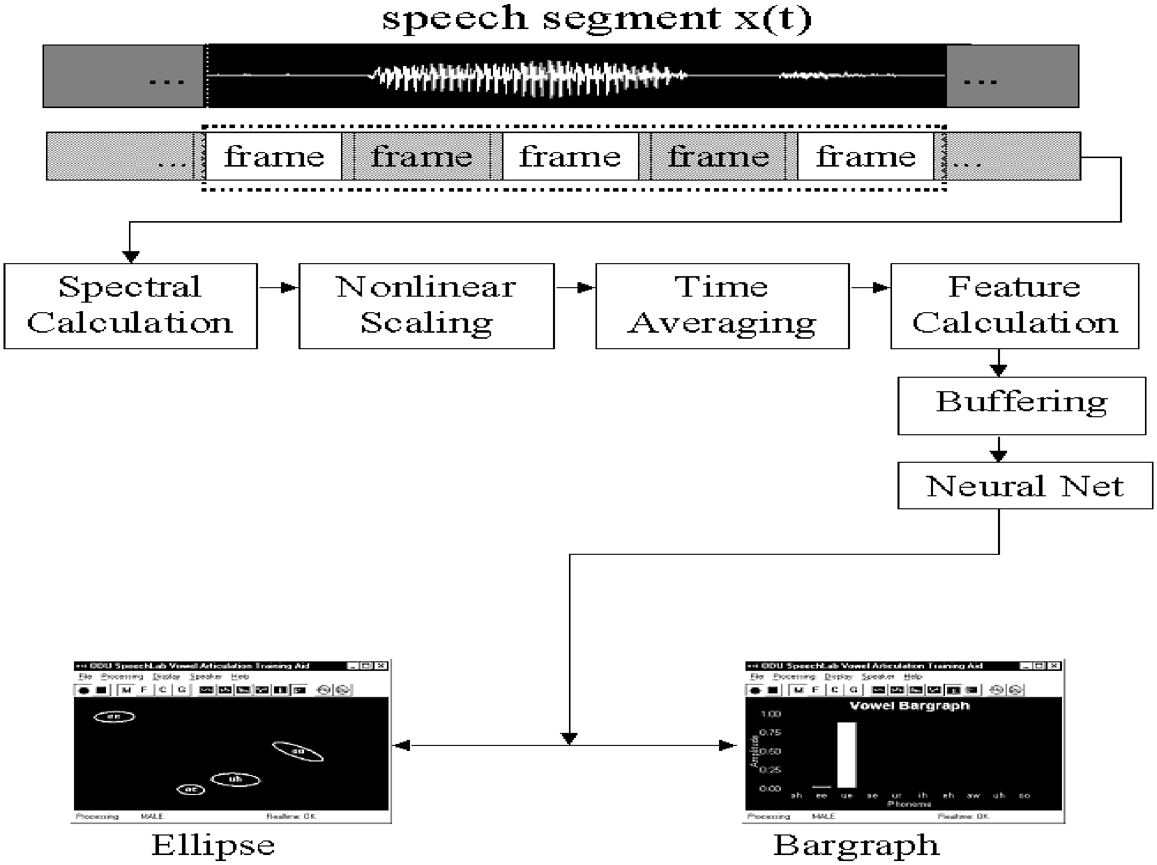

speech segment x(t)
...
...
...
frame
frame
frame
frame
frame
...
Spectral Calculation
Nonlinear Scaling
Time Averaging
Feature Calculation
Buffering
Neural Net
ODU SpeechLab Vowel Articulation Training Aid
File Processing Display Speaker Help
M F C G
ee
aa
uh
ae
Processing MALE Realtime: OK
ODU SpeechLab Vowel Articulation Training Aid
File Processing Display Speaker Help
M F C G
Vowel Bargraph
1.00
0.75
0.50
0.25
0.00
Amplitude
ah ee ue ae ur ih eh aw uh oo
Phoneme
Processing MALE Realtime: OK
Ellipse
Bargraph

Fig. 1. VATA signal processing block diagram.

Thesis Goals

The overall goal of the thesis is to improve the performance of VATA and to develop a system which extends the vowel displays to give feedback about the pronunciation of short words. The goals can be outlined as follows:

1) Improve VATA to increase the performance and its spectral stability.
2) To investigate a feature alternative to DCTC for VATA.
3) Develop Hidden Markov Models (HMM) and NN phone models to give feedback about the quality of short words.
4) Develop consonant vowel consonant (CVC) displays similar to the VATA to handle visual feedback of short words.

Chapter II details the improvements brought about in VATA with respect to its performance and stability. Chapter III gives a detailed account of the framework that has been developed to give feedback of short words (CVCs). Chapter IV lists the display processing steps and few displays that give visual feedback about the pronunciation quality of the CVCs. Finally, Chapter V concludes the thesis by summarizing the results and the improvements that can be brought about in the future.

CHAPTER II

GENERAL IMPROVEMENTS IN VATA

Introduction

One main requirement for the development of VATA is the availability of a “good” large database with a minimum of extraneous noise and mispronunciation. Previous work reported the effects of size of the training database on both the training and test recognition results when a significantly large amount of speech data is available [13]. Another important issue investigated by Zimmer was the spectral stability of long duration sustained vowels. This point was investigated since the spectral stability for VATA appeared to be low as compared to spectral stability for vowels displays with the previous system developed using a TMS320C floating-point DSP platform and custom-built preamplification circuitry. Zimmer especially looked at the effect of the various sound cards on spectral stability of the vowels. This chapter gives an account of some further improvements in VATA, especially with regard to improving the quality of the database and some improvements in spectral stability. We also report on an attempt to use a different front end feature set—spectral moments.

Database Pruning

The Speech Database consists primarily of vowels spoken by three categories of speakers (males, females and children). This data was recorded by means of a convenient user-interface program in a relatively quiet room. The vowels that were collected were /ah/ (/aa/), /ee/ (/iy/), /ue/ (/uw/), /ae/ (/ae/), /ur/ (/er/), /ih/ (/ih/), /eh/ (/eh/), /aw/ (/ao/), /uh/ (/ah/) and /oo/ (/uh/) which correspond to the vowel sounds found in the words “cot,” “beet,” “boot,” “bag,” “bird,” “pig,” “bed,” “dog,” “cup,” “book,” respectively. The first sets of each pair of vowel symbols are the Old Dominion University (ODU) phonetic vowel symbols, and the vowel symbols inside the parentheses are the ARPABET symbols. Up to the date of the writing of this thesis (June, 2003), a total of 14069 vowel tokens from 583 speakers (146 males, 171 females, 266 children) were collected (Table I). This data has then been used to train the neural-networks (NNs) for speaker-independent recognition of vowels, thus enabling several types of displays for VATA. Even though most of the mispronounced and noisy vowel tokens were eliminated at the time of the recording by the person monitoring the recordings, (typically an ODU graduate or undergraduate student), unfortunately a large number of “problem” tokens remained in the database. This "bad" data was assumed to be a major factor in degrading performance of vowel recognition for VATA.

TABLE I
Distribution of vowel tokens by categories before and after bad data removal

	No. of Speakers / No. of Recordings			
Token Description	**Male**	**Female**	**Child**	**Total**
Vowels before bad data removal	146/4760	171/5068	266/4241	583/14069
Vowels after bad data removal	146/4215	171/4421	266/3503	583/12139

Method of Removal

The approach of manually going through all of the recorded vowel tokens to determine the bad ones using visual and auditory inspection was first considered. However, although this approach was used for some of the database, it was not feasible to thoroughly inspect the entire database with this method. Rather, due to the large amount of data, an automated approach was needed. In particular, it was decided that a feed forward perceptron NN classifier trained to classify the vowels could be used to sort "good" vowels from "bad" vowels. The NN used, consisted of 1 hidden layer with 25 nodes. The inputs to the NN were the 12 scaled discrete-cosine transform coefficient (DCTC) features; the network was configured with 10 output nodes corresponding to the 10 vowels. The method of removal was implemented as follows. The classifier was trained on all the data for a particular speaker type (men, women, or children), and then used to classify all the training tokens. All tokens misclassified were noted and removed from the database. A total of about 12139 tokens remained (Table 1). The reduced set of tokens was then used for additional training of the NN classifier. This method did, of course, lead to almost 100% classification accuracy, since the tokens which were not classified correctly after the first phase of training (50,000 NN updates) were removed from the data. Note that the additional training (200,000 NN updates) was done because the resultant NN appeared to function somewhat better in the real-time system, but did result in a slight drop in classification performance on the training data. The vowel training recognition results obtained before and after bad data removal is shown in Table II.

The "correctness" of the procedure just described was verified in two ways. First, about 1000 tokens classified “bad” by the NN were listened to and about 85% of them were found to be either mispronounced or noisy. Secondly, NNs trained on the reduced data set appeared to result in better performance for the real-time VATA. As a result of these findings, it was decided that all the data classified “bad” by the NN would be removed completely from the database.

TABLE II
Vowel training recognition results before and after bad data removal

	Recognition	Percentages		
Token Description	**Male**	**Female**	**Child**	**General**
Vowels before bad data removal	88.55%	87.23%	82.59%	80.13%
Vowels after bad data removal	99.60%	99.71%	99.17%	99.47%

Spectral Stability in VATA

One assumption underlying VATA and in speech science in general is that the spectrum of a sustained vowel, as a long single utterance, is unchanging over time. Thus, the log magnitude spectrum of a vowel, as computed from consecutive frames each of duration approximately 30 ms, should be the same from frame to frame. Similarly, the DCTC features from these consecutive frames should be unchanging over time. This spectral stability for vowels appeared to be valid for the version of VATA developed with a DSP card. However, both the spectrum and resultant DCTC features appeared to have much larger variations over time in the PC version of VATA. The reason for the spectral instability in all-PC version of VATA was assumed by Zimmer [13] to be one of the following: (1) Program code errors introducing jitter in the input speech signal (2) Sound card quality contributing to the noise (3) Inherent jitter in the speech. Zimmer carefully examined the code, and experimentally compared performance with that obtained with comparable Matlab simulations, and decided that there were no problems in the code leading to the spectral instability. He also experimentally verified that only very low quality sound cards (such as the built in ones in a typical laptop) contributed significantly to spectral jitter. Zimmer concluded that the primary source of instability was due to inherent jitter in speech. Since this speech jitter, in terms of the spectrum computed from frame to frame, had long been recognized as a potential problem, time smoothing options had been incorporated in the code. The type of smoothing that was incorporated was peak smoothing, motivated by the idea that the peaks in vowel spectra are the most noise-free and also the most important for vowel perception. Nevertheless, even with the smoothing enabled over intervals exceeding 100 ms, considerable spectral and feature "jitter" remained in the display. Although impossible to confirm, it was thought that this "jitter" problem reduced the accuracy of the vowel displays.

As the name itself suggests, the peak-smoothing algorithm selects the maximum log-magnitude spectral value over a pre-specified number of time frames, over some interval ending in the current time frame in the time segment being processed. This particular number of time frames is a configurable parameter specified in the input file of VATA. This maximum spectral amplitude is initialized to the current time frame under consideration and this process is repeated for all the frequency components present. As the number of frames for time smoothing is increased, it was expected that the jitter in the spectrum would be reduced. However, it was noted that increasing the time smoothing window length did not really substantially reduce the spectral jitter.

This jitter issue was again examined in this study, and it was found that some of the initializations needed for smoothing after each speech segment is acquired had not been done and, therefore the smoothing was not done correctly. These initialization issues were corrected, and the smoothing was fixed. A new option for time smoothing, average smoothing was also introduced. It employs a similar procedure as for the peak smoothing, except the average of the spectral amplitudes over the specified number of time frames is computed for each frequency component. It was decided that for the real-time display system that average time smoothing option should be used as it takes into account the spectral components of each frame, over the specified window of time frames, rather than just the maximum over these frames, which, in extreme cases, can be due to noise. Several informal tests were also conducted with the real-time system comparing the two types of smoothing, and the average smoothing method appeared to result in somewhat higher accuracy.

Note that the smoothing window length is adaptively increased from 1 frame, up to the maximum number of frames specified, as a vowel is produced. Thus, near the beginning of each vowel production, there is very little smoothing, but the amount of smoothing increases if the vowel is sustained. This insures that silence frames are not averaged in with the vowel spectra, which would have the effect of flattening the spectra near the beginning of each production. To illustrate the effect of this smoothing, the mean and the standard deviation of the spectral amplitude for a steady state vowel of duration 3 seconds is shown for 3 levels of average time smoothing— 0 (no time smoothing), 10 frames (typical) and 100 time frames (Figs. 2,3). Note that the corresponding smoothing times are 0, 100 ms, and 1 second. As expected the standard deviation is greatly reduced with the time smoothing, especially as the smoothing window becomes very long.

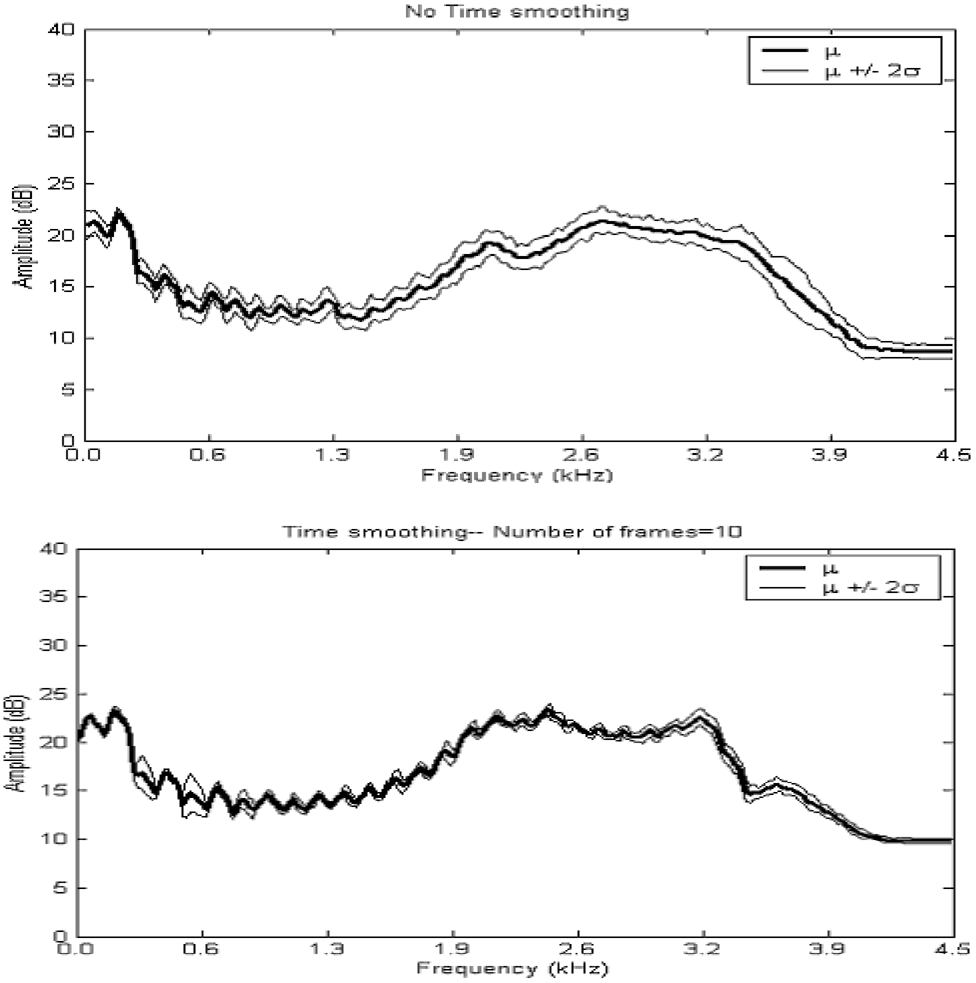


Fig. 2. Spectral stability for male speaker /ee/ token without time smoothing, with average time smoothing over 10 frames (100 ms).

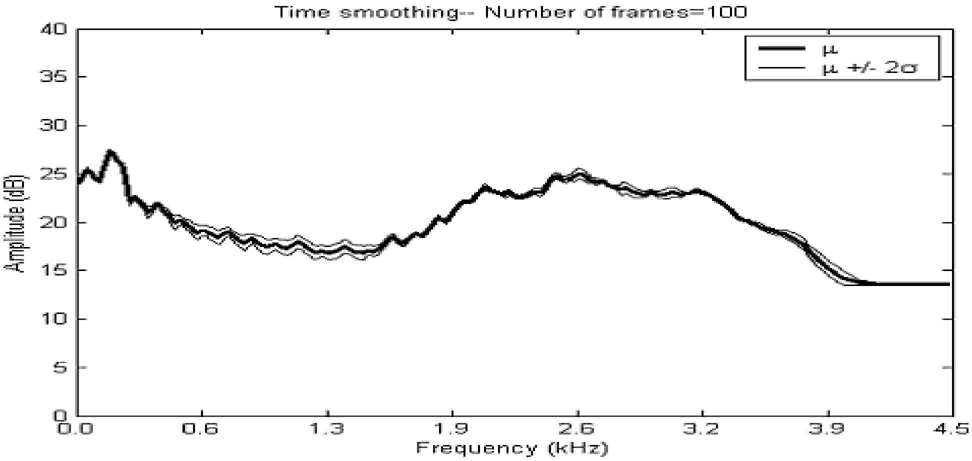


Fig. 3. Spectral stability for male speaker /ee/ with average time smoothing over 100 frames (1s).

Spectral Moments – Feature Alternative

DCTCs have been used as the features for the VATA system ever since the VATA was initially developed. Although DCTC features do seem to perform relatively well, and are very similar to the Mel cepstral coefficients used in most automatic speech recognition systems, it was still felt that there might be a better feature set. In this direction, "Spectral Moments" were examined as a possible alternative to the DCTCs as features. The formula used for the calculation of spectral moments is given by the following equation:

$$\text{SM (n)} = \sum_{i=0}^{i=N-1} (Xi - \overline{X})^n f(Xi), \; n = 12(typically) \tag{1}$$

f(Xi) in the equation represents the scaled log-scaled spectral component obtained by dividing the spectral component value by the sum of the spectral components for the particular time frame for which the features are being computed. $\overline{X}$ represents the mean (i.e.) the first spectral moment and *Xi* represents the various spectral components. *N* represents total number of frequency components available for the particular time frame. In total, about 12 spectral moments were calculated per time frame of the speech input signal.

Theory of Moments

Moment generating function of a discrete random variable x is given by the following equation:

$$\Gamma(\mathrm{z}) = \mathrm{E}(\mathrm{z}^{\mathrm{x}}) = \sum_{n=-\infty}^{+\infty} P(x = n)z^{n} = \sum_{n=-\infty}^{+\infty} P_{n} z^{n} \qquad (2)$$

If this equation is looked into more detail, it will be seen that it has a form very close to an ordinary z-transform. In fact, Γ (1/z) represents the z-transform of the sequence P_n= P(x=n) [14]. Upon differentiation, the above equation leads to a way of producing the various moments,

$$\Gamma^{(k)}(z) = E\{x(x-1)(x-2)...(x-k+1)z^{x-k}\} \qquad (3)$$

For z=1 and writing down the first two moments in the above equation gives

$$\Gamma^{'}(1) = E\{x\} \text{ and } \Gamma^{' \, '}(1) = E\{x^{2}\} - E\{x\} \qquad (4)$$

that reveals that moments are a form of autocorrelation and hence comparable to the DCTCs without warping.

Recognition Results

The Spectral moments were used as features to train the NN classifier in place of the DCTCs as explained previously. It was found that the spectral moments yielded poor results when compared to that when DCTCs (without warping) were used as speech features as shown in Table III.

TABLE III
Vowel training recognition results using spectral moments and DCTCs

Features	Recognition Percentage
Spectral Moments	59%
DCTC	90%

Spectral Moments Experiment Summary

The objective of the experiment was to find out if spectral moments were a good feature alternative to DCTCs. The theory of moment generating functions revealed that the spectral moments are a form of auto-correlation and are actually quite similar to the DCTCs without frequency warping. NN recognition results that were obtained with spectral moment features were considerably lower (by 30%) than results obtained with DCTCs, as presently implemented. Although it is likely that improvements could be made in the method for computing the moments, the theoretical analysis combined with the very poor experimental result, led us to conclude that spectral moments are not promising as features for recognizing vowels.

Sampling Frequency effects – Downsampling

The speech database consisting of vowels used for training the NN classifier was recorded using two sampling rates: 11.025 kHz and 22.050 kHz. Previous work studied the sampling frequency effects because of the use of 2 different sampling frequencies for training and testing [13]. It was reported that the test results were higher by 5% to 15% if the training and test sampling rates match versus having training data at one sampling rate, and the test data at a different sampling rate. (Note, however, the DCTC features were computed over the same frequency range for both training and test data). This implies that for “ideal” performance of the VATA system, the data used to train the NN should match the “real-time” mode’s sampling rate. Another important result of the tests done was that there was no appreciable gain obtained by using the sampling rate (for both training and testing) of 22.050 kHz over 11.025 kHz.

Method of Downsampling

As a result of the findings above, it was decided that all the tokens sampled at 22.050kHz would be downsampled by a factor of 2 to bring the sampling rate down to 11.025 kHz. Thus, the entire database would be effectively sampled at 11.025 kHz.

Figure 4 shows the steps involved prior to computing the DCTCs (features) for training the NN parameters. Downsampling (decimation) was achieved by averaging two samples of each vowel token to form the new sample. The transcription information that contains the location of the vowel information along with the start and end of speech is also updated to reflect the decimation. Note that this downsampling would not contribute to any loss of information, as the highest frequency component present in speech (at least for vowels) is less than 5 kHz and a sampling rate at twice this frequency is enough to completely represent the inherent information.

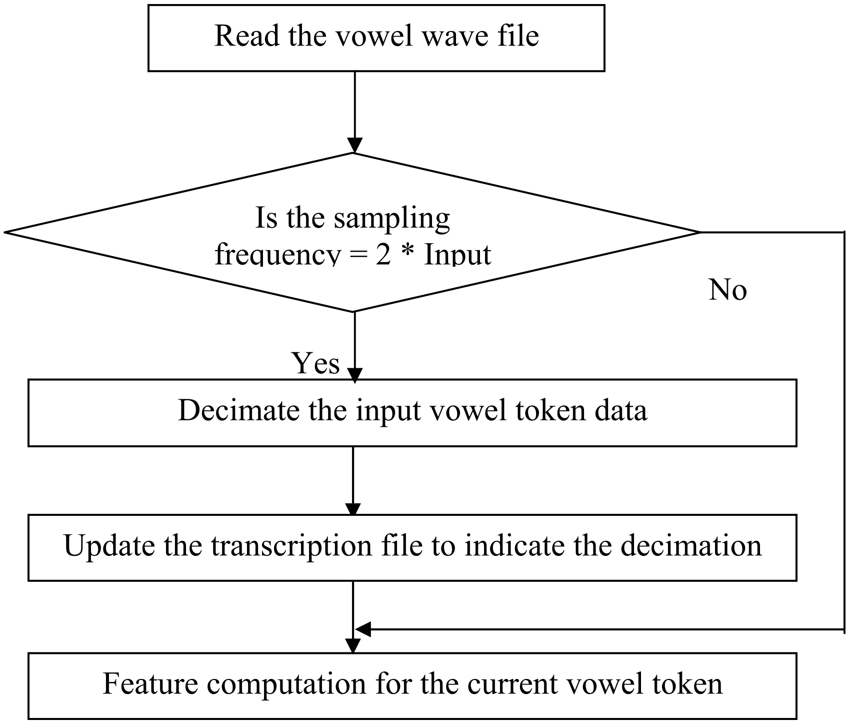

Read the vowel wave file
Is the sampling frequency = 2 * Input
No
Yes
Decimate the input vowel token data
Update the transcription file to indicate the decimation
Feature computation for the current vowel token

Fig. 4. Flowchart showing the steps involved prior to computation of features if decimation is involved.

The process of NN training was done as before with increased recognition results compared to the mixed-case (Table IV). Note that the results in the table are for the pruned database with "bad" tokens removed. The NN1 results are for the bargraph, and the NN2 results are for the ellipse display. The “real-time” display system input sampling rate was changed to 11.025 kHz to take advantage of the above process. It was observed that in line with the training recognition results, the real-time system’s performance also improved, particularly for the bargraph display.

TABLE IV
Vowel training recognition results with and without downsampling

	Male (%)		Female (%)		Child (%)		All (%)	
	NN1	NN2	NN1	NN2	NN1	NN2	NN1	NN2
1.No Downsampling	99.88	98.07	99.86	96.98	99.77	95.92	99.39	94.35
2.With Downsampling	99.95	98.2	99.89	97.45	99.82	96.18	99.49	94.45

Summary

A summary of the work presented in this chapter, and the main conclusions of that work, are as follows:

(1) For building any effective NN classifier "good" data is required. Pruning of the vowel database by removal of the bad data consisting of mispronounced and noisy speech data resulted in improved performance of the NN training results and also operation of the "real-time" (VATA) system.

(2) The issue of spectral instability in VATA was "solved" using an adaptive length time-smoothing algorithm based on either peak smoothing or average smoothing. This smoothing greatly reduced observed jitter in the spectrum. Average smoothing appears to be somewhat better than peak smoothing, and thus was incorporated into the real-time display.

(3) An attempt at using Spectral Moments as an improved feature set in place of the existing DCTCs was not successful. A theoretical examination of the Spectral Moments indicated that the moments are actually a form of autocorrelation coefficients and thus very similar to the DCTCs without frequency warping. Also, the recognition results obtained with spectral moments were very low.

(4) Extending the previous work done, all the tokens sampled at 22.050 kHz were downsampled by a factor of 2. Though this resulted in negligible improvement in the training recognition results, it did improve the performance of the real-time system.

The work summarized (except for the spectral moments efforts) resulted in improvements of the real-time system. Even though the visual speech display for the vowels has been improved to a large extent, an unavoidable problem still remains. Isolated vowel pronunciation is relatively difficult (compared to isolated words) and unintuitive. Also, the systems capability of improving the speech of hearing disabled children will be vastly improved if the speech display could be extended to CVCs (Consonant Vowel Consonant words) in addition to the vowels. The following chapter describes the efforts made in this regard. It describes in detail the process chosen for phone level labeling of the CVCs (words). Chapter IV details the displays of the extended system for CVCs.

CHAPTER III

HMM-BASED TRIPHONE MODELS FOR SEGMENTATION AND LABELING OF CVC DATABASE

Introduction

In Chapters I and II, the signal processing steps and display options have been described for giving real-time feedback about the quality of pronunciation for 10 steady-state American English monopthong vowels (/aa/, /iy/, /uw/, /ae/, /er/, /ih/, /eh/, /ao/, /ah/, and /uh/). This vowel training aid is thus referred to as a Vowel Articulation Training Aid (VATA). The previous chapter explained in detail the improvements brought about in VATA. The non-naturalness and clear limited scope of vowels produced in isolation led to the exploration of a computer-based articulation training of CVCs (consonant-vowel-consonant). In this chapter, methods are described to develop a triphone-based Hidden Markov Model (HMM)/Neural Network (NN) recognizer such that real-time visual feedback can be given about the quality of pronunciation of short words and phrases [15]. Experimental results are reported which indicate a high degree of accuracy for labeling and segmenting the CVC database developed for "training" the display.

The objective of a CVC display is 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. Also, it is desired to provide feedback about the produced word in a variety of formats including game displays. In order to achieve these objectives for the display of phonetic information, it was first necessary to record and prepare a large database of CVC tokens. Preparation includes elimination of poorly pronounced tokens, determination of time markers for phonetic segments in the properly produced segments, and finally creation of HMM and NN phone models for each phone of interest.

Since it was desired that the system provide phone correctness, it would not suffice if just the HMM word recognition was done. As a result, NN phone models had to be built so as to implement real-time NN recognition at the phone level. This motivates the need for accurate phone-level segmentation and labeling of words.

Database of CVCs

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 does appear as a momentary silence and should be labeled by the labeling algorithm. Note, however, that this 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 V 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.

TABLE V
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 aa 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		

A database of CVC sounds was collected from adult males (145 speakers), adult females (166 speakers), and children between the ages of 6 and 13 (252 speakers) as listed in Table VI by means of a convenient user-interface program in a relatively quiet room. All tokens were automatically endpointed, using an endpoint routine similar to the Evangelos [16] algorithm for endpoint detection for isolated word recognition. Listeners, typically an ODU graduate or undergraduate student, then evaluated tokens, and those that appeared to be incorrectly pronounced were eliminated from the database. This data was collected over a period of several years during which the sampling rate was increased from 11025Hz to 22050 Hz. For all experimental results reported in this thesis, the 22 kHz data was smoothed with a 2-point smoothing window, and decimated to 11 kHz as explained in the previous chapter.

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.

TABLE VI
CVC database

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

Segmentation and Labeling of CVC Database

After considerable experimentation, manual examination, and testing, the following multi-step approach was used for segmentation of the CVC database.

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 agreed to be “incorrect” 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.

Each of these steps is now described in somewhat more detail below and illustrated with experimental data.

Step 1 Heuristic Rule-Based Segmentation

The onset and offset of speech (i.e., the endpoints) are first determined using an endpointing routine similar to Evangelos [16] algorithm for isolated word recognition. Next, a pitch track is computed for each utterance, using the pitch routine of Kasi and Zahorian [17]. Additionally three normalized spectral band energies were computed—low-frequency energy band (300-800 Hz), mid-frequency energy band (800-2000 Hz) and high-frequency energy band (2000-4000Hz). The vowel region of each syllable is located using a combination of the voiced portion of the pitch track, the normalized low-frequency band energy, and heuristic rules (including maximum and minimum durations for each initial and final stop). Then, the spectral band with the maximum value between the vowel end and the final endpoint of the CVC from all the 3 bands is selected as the band to be used to determine the starting point for the final consonant. The first point in time at which this normalized band energy exceeds an empirically determined threshold (typically .3) is selected as the beginning of the final consonant. In addition, if this energy band falls below another threshold (typically .1) for at least a certain time duration, in the region between the end of vowel and the beginning of the final consonant, then that low-energy region is labeled as closure.

Figures 4 and 5 illustrate this heuristic labeling method for two CVCs. Note that the method appears to give correct segmentation for the CVC depicted in fig. 5 whereas the final stop consonant appears to be in error in fig. 6. Visual inspection of several hundred tokens indicated that this heuristic segmentation appeared to be generally correct (no really large errors) for about 80% of tokens, but did have some really large errors for the remaining 20% of tokens. Thus, the heuristic method was not considered accurate enough for a "final" pass of segmentation, but was valuable for bootstrapping HMM models. The HMM models, followed by forced alignment, (described below) did appear to provide more accurate segmentation.

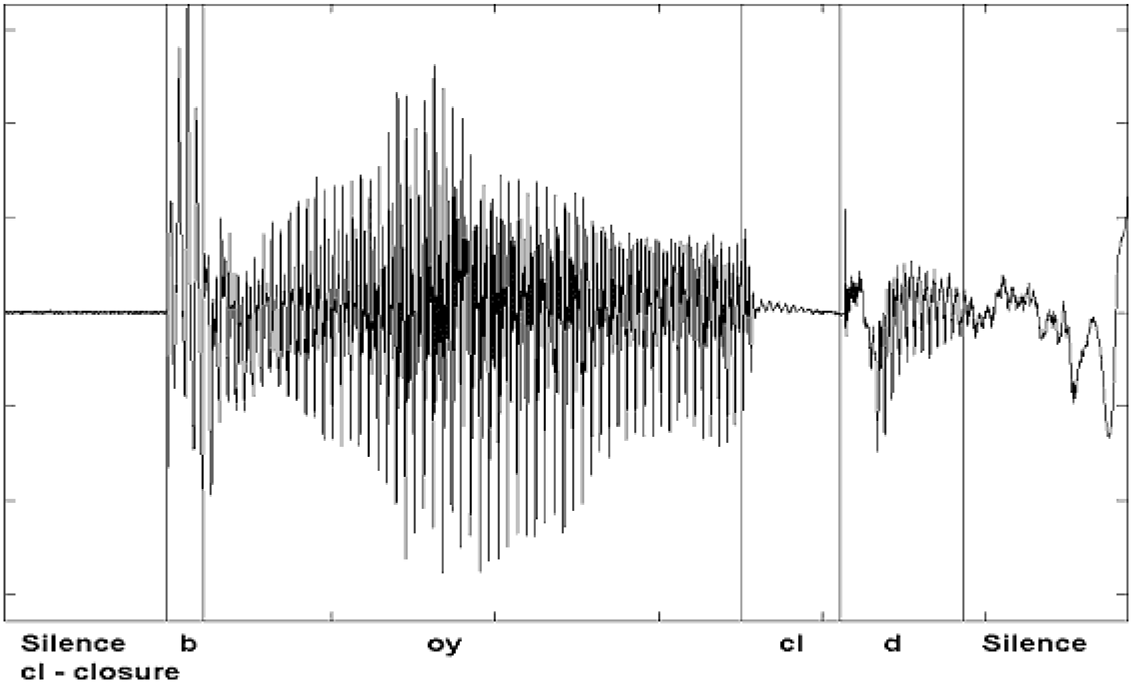


Fig. 5. Example of correct heuristic segmentation for “boyd”.

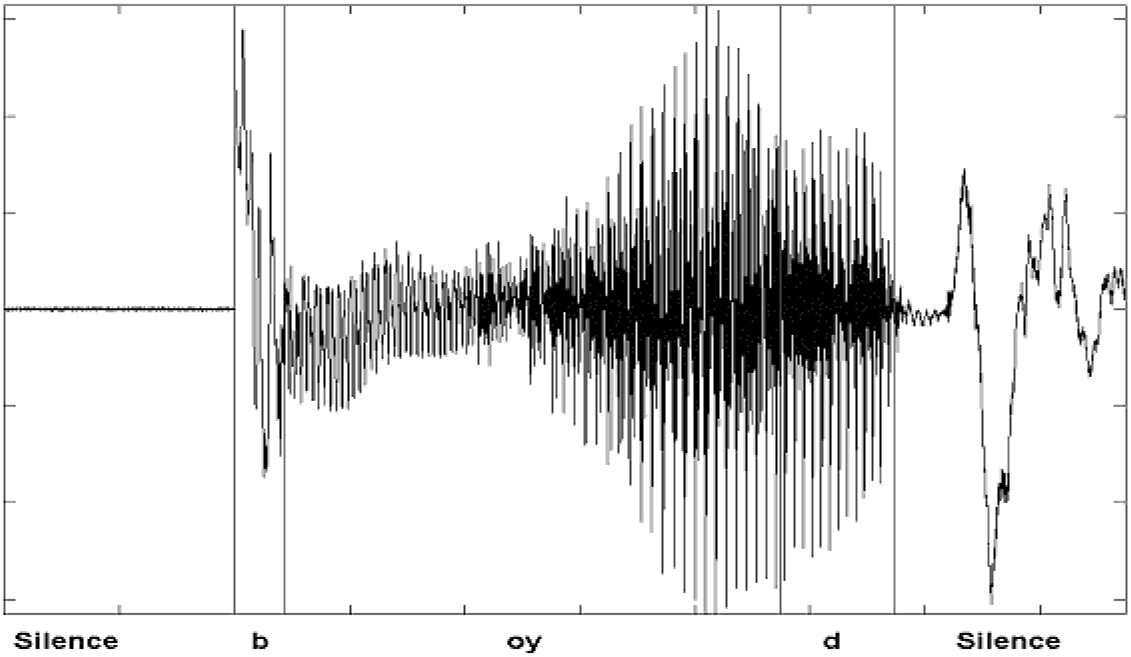


Fig. 6. Example of incorrect heuristic segmentation for "boyd"(final stop problem).

Step 2 Bootstrap triphone models

The heuristic labeling process just explained was used for initializing HMM triphone models for the CVCs. Initialization based on “flat-start” data (no segmentation used) resulted in lower recognition performance than initialization based on the heuristically labeled data. The HTK (Hidden Markov model Tool Kit ver 3.1, [18]) was used in this process to build a recognizer, which was used to determine the degree to which the words present in the training database could be automatically recognized. The HTK was configured with a dictionary which allowed the presence or absence of closure in each CVC. Mel frequency cepstral coefficients (MFCCs) derived from the FFT-based log spectra were used as the features, augmented by delta and acceleration coefficients (totally 39 terms).

The HMM monophone prototype models were configured with 5 states, 39 features, and 10 Gaussian mixtures. This number of Gaussian mixtures was determined by iteratively increasing the number of mixtures from 1 through 10, and observing that the best word recognition results were obtained with 10 mixtures. This is understandable since 10 Gaussian mixtures implies that each probability density functions is modeled as a weighted sum of 10 Gaussian densities, each characterized by a mean and variance. Thus, the training process for the case of 10 Gaussian mixture implies that 10 means, 10 variances, and 10 mixture weights (30 total parameters) must be determined for each state in each HMM model. Note that even this number is very small compared to the number of parameters that would have been needed with full covariance matrix models for each state. In any case, the 10 mixtures per each feature leads to the feature space being increased and better separation of the phonemes in the feature space as compared to the case of having just 1 Gaussian mixture. A larger number of mixtures were not used, due to the large number of parameters needed, and the possibility that the training database was really not large enough to adequately train for a higher number of mixtures. After the prototype models were established, each monophone HMM was initialized by the HTK tool (HINIT), which uses the Viterbi algorithm to find the most likely state sequence corresponding to each training sample. Note that a total of 21 monophone models were created (13 vowels, 6 consonants, 1 silence and 1 closure model) as shown in Table VII.

TABLE VII
Incomplete list of monophone models (silence and closure not shown) and CVCs which contain them

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

These isolated-unit models were refined using the Baum-Welch re-estimation procedure (HREST tool in HTK). As isolated unit training is not sufficient for building phone models, embedded training (HEREST), which simultaneously updates all the HMM's in a system using all of the training data, was used. This was repeated for three passes. Finally, triphone models were built with state tying and using reestimation (typically 3 times). Example of the triphones that were built is shown in Table VIII.

TABLE VIII
List showing some of triphone models and CVC examples which contain them

triphones	Example	triphones	Example
b+ae	Bag	ih-g	Pig
b-ae+g	Bag	p-ih+g	Pig
ae-g	Bag	k+aa	Cot
b-ae	Bag	k-aa	Cot
g	Bag	k-aa+t	Cot
k+ey	Cake	aa-t	Cot
k-ey	Cake	b-er	Bird
k-ey+k	Cake	b+er	Bird
ey-k	Cake	b-er+d	Bird
		er-d	Bird

Note that some biphones are also present as a result of the word boundary symbol (silence). This form of notation is known as “word internal”. In this notation a “+” is used to add a monophone to the right and a “-“ is used to add a monophone to the left. Thus “b-ae” represents a left biphone and “b+ae” represents a right biphone and “b-ae+g” represents a triphone. The triphone models also have the same configuration as the monophone models, that is 5 states, 39 features and 10 Gaussian mixtures. Totally there were 71 triphone models generated.

Total processing time needed for all the above processes including initialization and re-estimation was approximately 45 minutes on a dual 2.38GHz xeon processor machine with 512MB ram. Summarizing again, there were 21 monophone models, 71 triphone models and the CVC database contains 16535 CVCs (.wav and .phn files).

Step 3 Bad data removal

Even though it was pointed out that listeners removed incorrectly pronounced and noisy tokens at the time of recording the tokens, there still remained "bad" tokens. Due to the tediousness and time required to listen to all tokens, an automated method was selected to identify and then remove the problem tokens from the database. In particular, the HMM models trained in the previous step were used as a recognizer. Those tokens not correctly recognized by the recognizer were then considered as potentially bad. Note that the database includes over 500 speakers, including adult males, adult females, and children, thus making the recognition task somewhat difficult.

As mentioned previously in the bootstrapping section, the number of Gaussian mixtures was iterated from 1 to 10 and for each iteration the “bad” tokens were identified. Cumulatively from all these iterations, 14% (2400 tokens) of the data was classified by the HTK recognizer as bad by this step. In order to verify the “correctness” of the above procedure, all those tokens classified “bad” were listened to. Only these tokens which listeners also agreed were bad were removed from the database. Table IX gives a list of all the tokens remaining after this step. The goal was to be confident that nearly all tokens used for final training were clearly and correctly produced.

TABLE IX
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

Note that about 70% (1677) of the tokens considered as potentially bad based on the HMM recognizer were also identified as bad by the listeners. Since the database was relatively large, it was not considered a major problem if some good tokens were also deleted. In any case, the original version of the database has also been saved, in case future techniques are better able to determine these "bad" tokens.

Step 4 Triphone model retraining

Triphone models were recreated using the identical processing methods as described above for step 2, except now, only data remaining after step 3 was used. Thus, it was expected that these HMM models would be better representations of the phones than those created in step 2, because of the elimination of the “bad” data. As a test, these triphone models were again used as a word recognizer, and approximately 99% accuracy was obtained on the training data.

Step 5 Final forced alignment

Finally, the above retrained HMM models were used to obtain a forced Viterbi alignment, resulting in final labeled training data. Some of the results of the labeling done by this method are shown in figs. 7 and 8.

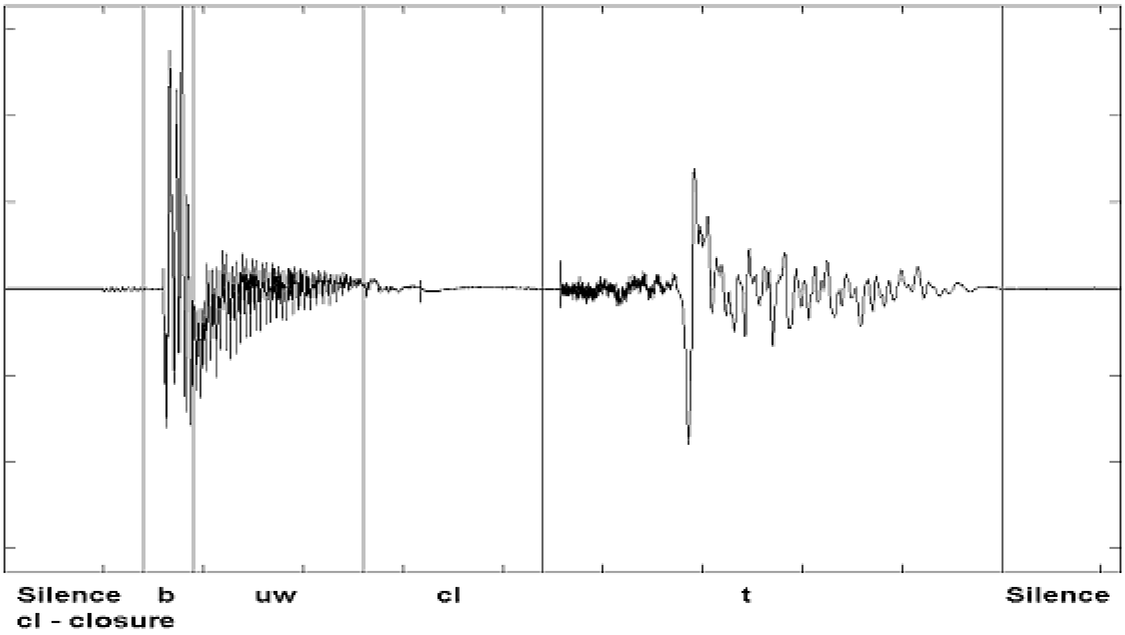


Fig. 7. Example of force-aligned segmentation for "boot".

In fig. 7 "cl" denotes the closure which corresponds to brief period of silence between the utterances of the vowel and the final consonant. Manual inspection of about one-hundred tokens did not reveal any major labeling errors which is an indication that the HMM phone labeling is superior to the heuristic labeling, and hopefully would also allow more accurate automatic recognition. Recognition, from a NN recognizer, using both methods for labeling is shown in fig. 8.

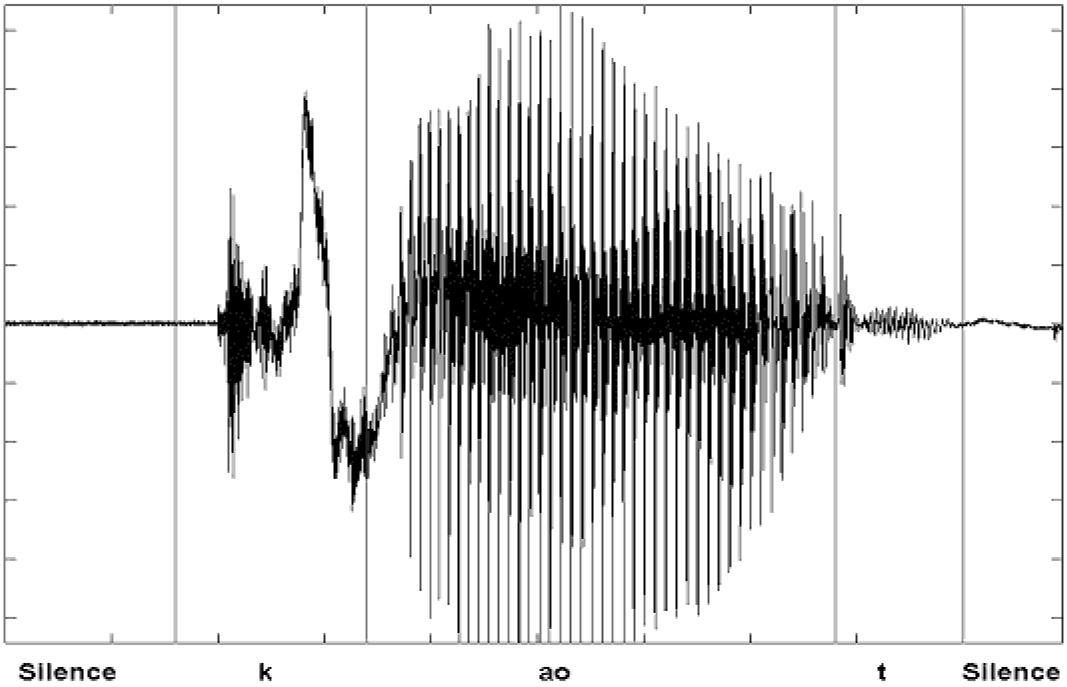


Fig. 8. Example of force-aligned segmentation for "cot".

NN Training

Three NNs were trained for possible use in the CVC display system—one each for the initial consonant, vowel and the final consonant. Each NN has one input node per feature component, 25 nodes in the hidden layer, and one output node for each phone to be recognized. The initial consonant network thus has 4 output nodes, the vowel network has 13 output nodes and the final consonant network has 5 output nodes. The NNs were trained using error backpropagation for (typically) 250,000 iterations. These NN classifiers resulted in the following recognition rates for the training data as shown in Table X. The recognition rates for the vowels are higher (7-10%) on the data obtained from the final forced alignment, rather than the heuristic algorithm alignment. However, for the initial consonants as well as the final consonants, the recognition results were approximately the same for the two labeling methods.

TABLE X
Percent correct results for NN classification of the phones in CVC syllables based on either heuristic labeling of phonetic segments or HMM forced alignment

	Initial Consonant	Vowel	Final Consonant
Heuristic Algorithm	87.27%	73.88%	78.12%
HMM/NN Based	87.00%	80.85%	79.66%

Summary

The extension of the Vowel Articulation Training Aid (VATA) for use with short words (CVCs) has been described. In response to audio input from a user, the goal is to provide a visual display provide feedback about the quality of pronunciation of the phones in each CVC. The requirements for this display are:

(1) Good Database with minimum of noise and mispronunciation.
(2) Good Labeled phone transcriptions.
(3) Good models for the phones.

A step-by-step procedure for segmentation and labeling of the CVC database had been explained in detail. NN recognition shows the improved phone recognition results over the heuristic algorithm. In the next chapter, training issues for the CVC displays, and the displays that have been implemented for the CVCs, are explained.

CHAPTER IV

CVC DISPLAY SYSTEM

Introduction

As mentioned in Chapter III, the limitations of a speech display that functions only for vowels produced in isolation led to the exploration of computer-based articulation training for Consonant-Vowel-Consonants (CVCs.) Although still far from "natural" speech, the CVCs are at least a step in the direction of more natural speech. The previous chapter explained in detail the Hidden-Markov Model (HMM)-based triphone-modeling of the phonemes present in the CVC and the use of the HMM for accurate labeling of the phonetic segments in each CVC. As will become clearer later in this chapter, this phonetic labeling step is critical in the development of a display of the phonetic information in CVCs. In this chapter, methods are described to develop a real-time visual display system giving feedback about the quality of the pronunciation of short words using a monophone-based Neural Network (NN) classifier.

CVC Display Processing Steps

The objective of a CVC display is to visually present indicators of pronunciation “correctness” at the phone level in real-time, in response to short words produced by a speaker. 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 the signal processing can be explained in terms of three levels of buffering (fig. 9). First, as mentioned above, the non-overlapping but continuous “segment” buffers form the interface between the data acquisition subsystem and the signal processing and recognition software.

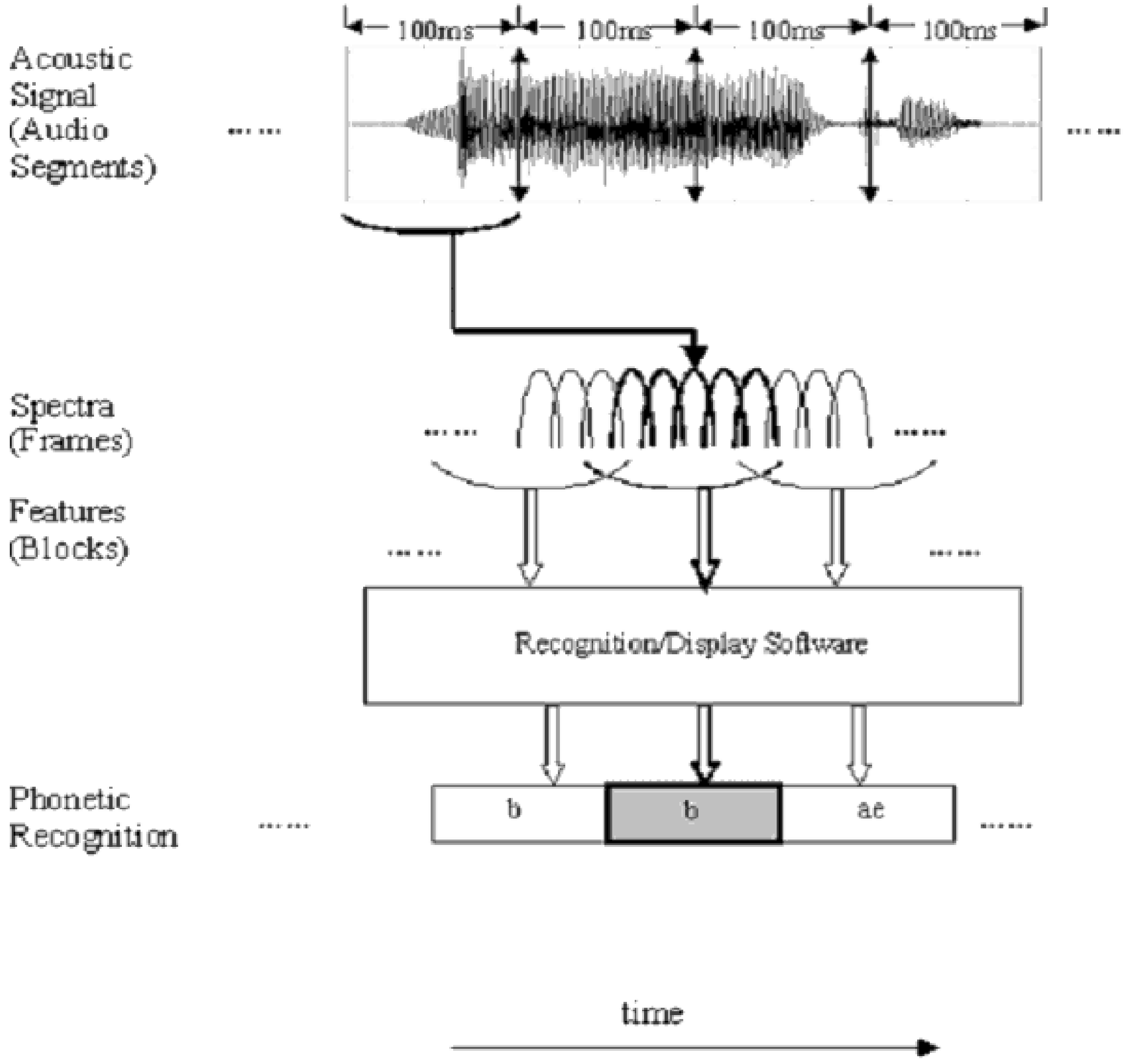


Fig. 9. CVC display block diagram.

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 consists of "blocks," with each block consisting of an integer number of frames, with block spacing also equal to an integer number of frames. All parameters ("features") used for recognition are computed from all the frames in a block with parameter updates based on the block spacing. Similarly, recognizer decisions are based on the current (and possible previous) block output parameters. For example, as will be explained in detail later in this chapter, the present version of the real-time system "computes" the top scoring phones for each block, as based on the parameters of that block. It is anticipated that future versions of the system will also use information from previous blocks. Display updates are made once per each new segment, using all the block-based recognizer decisions which occur in that segment. In particular, average of the NN outputs are computed by dividing the sum of all the NN outputs for each block present in the segment by the total number of blocks for each phoneme category and the one that has the maximum value is then chosen as the phoneme category. Each of the processing and recognition modules maintain 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 50 ms for the segment buffer, 20 ms frames with a frame spacing of 10 ms, and blocks consisting of 3 frames with a spacing of 1 frame between blocks. The time delay between input speech and display changes is on the order of one segment time, or 50 ms for the values given.

Note that the very low latency aspect of the real-time system is likely not needed, at least for the CVC displays currently implemented. That is, an approach could have been developed whereby the entire word was captured, analyzed, and then results displayed. The user would then see the results of the word, only after the word was completely uttered—or typically a delay of perhaps 1 second. This "whole word" approach was not used, since it would not seem to extend as well for the display of speech information contained in entire sentences spoken spontaneously. For example, most present day commercial speech recognition dictation systems are based on a whole word approach, and typically introduce delays of several seconds extending over several words. In contrast, the approach outlined above can be extended to work with continuous speech, with very short delays between a user production and display changes. For the purposes of speech training, this lower latency would appear to be very important so that the user can immediately see the "correctness" level of his/her productions and take corrective actions as needed.

CVC Signal Processing Steps

The signal processing itself is illustrated in fig. 10. Primary steps consist of mid-frequency pre-emphasis, windowing, spectral magnitude calculations, discrete cosine transform coefficient (DCTC) computations for each spectral frame, discrete cosine series (DCS) calculations over the DCTCs in each block, scaling, and finally NN classifications.

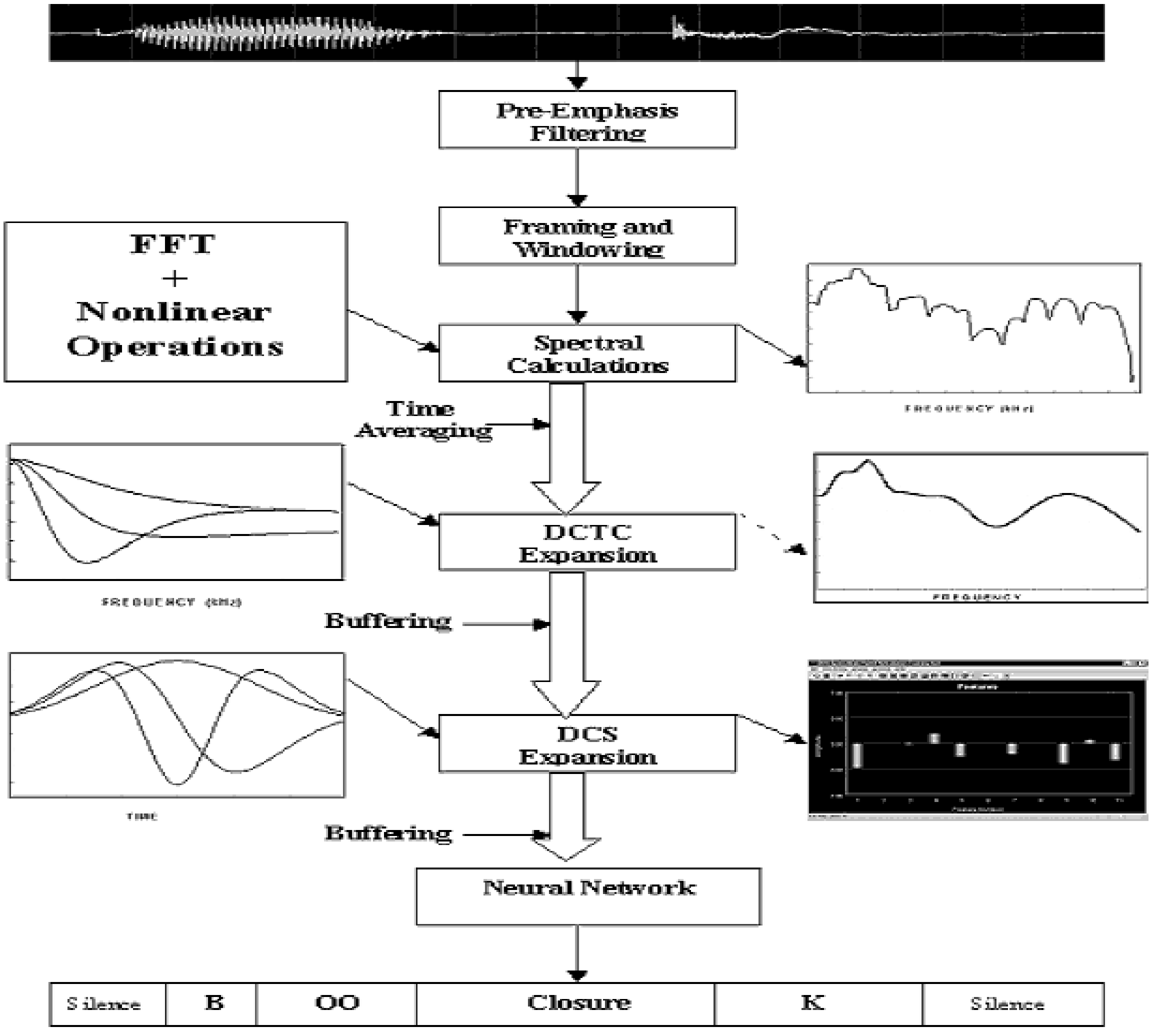


Fig. 10. CVC signal processing block diagram.

More details of this signal processing are given by Zahorian [19]. However, for convenience to the reader, the primary steps are summarized here. The primary spectral parameters, DCTCs, similar to cepstral coefficients, are computed using cosine basis vectors that are modified so that the frequency resolution approximates a Mel frequency scale. The DCTCs 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. The coefficients of this second transform are called Discrete Cosine Series Coefficients (DCSCs). This method is very flexible with a small number of parameters that control the details of the analysis, particularly in terms of spectral/temporal frequency resolution tradeoffs. Currently, 12 DCTC terms are computed, followed by 3 DCS terms for each DCTC (36 total parameters). The NN classifier is a feed-forward multi-layer perceptron that attempts to classify the current input (that is scaled DCS terms for the current block) in terms of one of 20 phonetic categories (Actually 21 phonemes, but for convenience purposes, closure was also classified as silence).

In order to implement the system just described, a NN is trained as a phonetic recognizer, based on a large database of correctly produced CVCs. The database in use consists of 13 CVC words, containing 20 phones. The recognition module returns the identity of the top 3 scoring phones for each segment as explained in the display processing section. The goal is to use this information to provide feedback about the produced word in a variety of formats including game displays. In the following section, NN training issues are described.

NN Training Considerations

During the training phase, the NN requires a “target” label at each point in time (block for this case) for which a phonetic decision is to be made. This is in addition to the features (DCSCs) calculated for every block. Additionally, for real-time causal operations, decisions can be based only on present and previous blocks. In order to perform NN training in the above manner, a prime requisite is the availability of detailed phonetic labeling as discussed in the previous chapter. Additionally, each block should then be assigned a target corresponding to the phone category in the time interval in which the block is primarily located. The issue of assigning phonetic labels to blocks is made more complex due to the asynchronous relationship between blocks and phonetic segments. For example, a block may span 2 or more phones. Some phones may be much shorter than a single block, especially initial consonants.

In consideration of the issues mentioned above, the front end analysis software that has been in use in the speech lab for several years has been written to be compatible with the acoustic ("wav") and label information ("phn") files, and allows a variety of methods for selecting and sorting desired phonetic segments from each file in a database. In particular, phonetic segments are selected (and later processed) using a parameter known as the frame selection method (FSM) in the front-end feature calculation routine. FSM=2 is used to select the entire phonetic interval as given by the labels (as explained in the previous chapter), but as modified by "start" time and "stop" time parameters, which can be used to modify the labeled interval for each phone. For example, the front end analysis program could be configured to select each labeled phonetic segment (for the phone specified), beginning 50 ms before the onset label of the phone, and ending 25 ms after the labeling ending time for the phone, by letting FSM = 2, start time = -50, stop time = 25. Alternatively, FSM=4 is used to select phonetic segments, with respect to the labeled midpoint of each segment. For this case, the start time is used to specify the length of the analysis time window prior to the midpoint and the stop time is used to specify the length of the time window size after the midpoint.

One primary advantage of signal processing based on the approach mentioned above, is that the method is already available and it is compatible with existing NN training programs. For example, the front end analysis program creates one separate feature file for each phone. Each feature file then contains one feature vector for each block of each selected segment for the phone corresponding to the that file. In effect, each feature file stores a large matrix, containing all the training data for a particular phone category. The NN training program can then be used to attempt to classify the data in these files, with the assumption that data in each file corresponds to a different category. The alignment problem between the blocks and the phonetic segments as mentioned before is also solved. For all these reasons, the procedure just outlined was used for some testing, as described below.

However, this procedure does have some limitations and possible drawbacks. For example, a disadvantage of using FSM=2 is that very short stop consonants like ‘b’,’d’,’p’ will not be used at all from many of the utterances. That is, for cases where selected phonetic segments are shorter than a block, they are ignored and hence will not be taken into consideration for feature computation. However, FSM=4 can be configured such that 1 block is selected for each phonetic segment with the block centered at the labeled midpoint. This is done by setting the block length equal to the phonetic segment length, as determined by the configurable start and stop time.

The major perceived disadvantage of both these segment selection methods is that they exclude the possibility of examining previous blocks in addition to the present block while making phonetic decisions. The use of present and past blocks is the underlying principle behind a time-delay neural network (TDNN) to take advantage of the chronological history inside the CVC utterance. There is considerable evidence in speech science which indicates that much more reliable phonetic decisions can be made if coarticulation effects are taken into account, which implies that such a history is needed.

Therefore, a different approach for managing data in each utterance was used to preserve the temporal history within each utterance, and also to handle issues involved with very short phones. In particular, at the front end analysis stage, the entire CVC utterance is processed as one chunk and the phone labeling information is not used at all at this stage (FSM = 0 in the analysis program). This procedure is repeated for the entire CVC database and the result is a separate feature file for each utterance in the database. Each utterance feature contains feature vectors corresponding to total number of blocks in that utterance. Note that for this FSM, the start and stop times for this method refer to the ends of the CVC utterance.

In order to accommodate the type of processing mentioned in the previous paragraph, and also manage and take advantage of the temporal history, the data management in the NN training program had to be completely changed, and one major new routine written. First, the NN routine reads all the utterance feature files, preserving the temporal order within each file, into one large two dimensional array. The routine also reads in all the original phone files, thus allowing the correspondence between feature vectors and phone labels (or "targets" for NN training) to be determined. These phoneme targets are determined by a newly written routine that determines the phones present in each block of each CVC utterance using this phone labeling information for that utterance. The phone that accounts for the major portion of the block is chosen as its target category. This procedure is repeated for all of the blocks present in the rest of the CVC utterances in the database. One difficulty is that in some cases very short phones (particularly the initial consonants) might not be assigned to any block in the utterance. Therefore, a "correction" algorithm was developed that ensures that each phoneme present in the CVC is assigned as a target category to at least one block. For example, in the case of absence of the initial consonant, one of the vowel blocks at the start of the vowel of the CVC is changed such that the block is assigned to the initial consonant that was previously absent.

This phone block labeling method is applied to the entire database, such that a feature vector and target label is available for each block in the database. With this approach, the NN was configured to be trained as a classifier for each block, using the target for that block, and a specified number of previous blocks as features. At CVC utterance boundaries, the previous block feature inputs to the NN are filled with zeros, since there is no valid history present at the beginning of each utterance. An additional modification to the NN training was that it was found necessary to randomize the selection of blocks used for training, rather than to select training blocks consecutively. The reason for this is presumably due to the inappropriateness of using several consecutive blocks from the same phone (for long vowels), causing the NN to over adjust for that particular phone.

Experimental validation and Training Results

NNs were trained for the CVC display, using each of the methods mentioned above, and the entire labeled CVC database mentioned in Chapter III. Summarizing briefly, the database consists of 13 unique CVCs consisting of 20 phonemes collected from adult males, adult females and children between the ages of 6 and 13. The CVC database size was close to 15,000 tokens after the HMM labeling procedure was completed as explained in the previous chapter.

As in the case of the vowel display system (Chapter I), DCTCs were used as the frame-level features. Typically 12 DCTC were computed for each frame. These 12 DCTCs were then each encoded by 3 DCS terms to encode changes over time, thus resulting in 36 features per frame. Typical values of some of the parameters used for this feature extraction are 20ms frame size, 10ms frame spacing, 3 frames in each block and 1 frame block spacing. Tests were conducted for each of the frame selection methods mentioned above, (FSM values of 0,2 and 4.) As mentioned previously for the case of FSM=0, the number of the feature files is equal to the number of CVC utterances (files) in the database and for the other 2 cases (FSM = 2 or 4) the number of feature files is equal to the number of phones (20 for this current CVC database—for convenience purposes, closure was also classified as silence).

The NN used in the CVC display system is a feed-forward multi-layer perceptron with one hidden layer which attempts to classify each block of each utterance as one of the 20 phonemes. NNs were trained using each of the methods mentioned for selecting phonetic segments. In order to compare performance, two types of evaluations were made, as follows

1) phone recognition rates were compared for the training database.
2) Informal human evaluations were done on the real-time system for each trained NN.

The first kind of NN training was performed on the same NN classifier that was used in the vowel-display system, but with modified front end feature calculations using parameters as shown in the following table (Table XI). FSM values of 2 and 4 were used for this training. The NN classifier used for these cases typically had 36 input nodes, 25 hidden nodes and 20 output nodes.

For the other type of training (FSM = 0), the NN was modified, as mentioned above to take advantage of temporal history, using a TDNN. Thus, this NN bases its decision not only on features in the current block but also that of the previous blocks. This number of the previous blocks is configurable and it is typically 3. Typically, this NN has 108 (12*3*3) input nodes, 25 hidden nodes and 20 output nodes corresponding to the 20 phones. Note that the newly labeled database as explained in the previous chapter was used for this training and the labels were used for specifying the targets separately to the NN, as explained above.

The NN recognition training results are reported in the following table (Table 11). The recognition results obtained for the case of FSM=4 gives the highest recognition rate of about 75%. However, it also should be noted that the total number of tokens (blocks) selected (roughly 4 times the number of CVC recordings (14858) with one block each for each of the phones in the CVC i.e. silence, initial consonant, vowel and the final consonant) is considerably fewer than for FSM=0 or 2. For the TDNN case (FSM=0), the figures show that the recognition results improve as the number of blocks is increased from 1 to 5. Also note that the CVC database had different classes of speakers — males, females and children, all of which were trained as a single group. Presumably higher rates could have been obtained if separate networks had been trained for each group. The biggest issue with all of these results is that they are based on training data only, and thus performance on new data ("test" data) cannot be determined.

TABLE XI
NN training recognition results of the phones (20) present in CVC database

Recognition % / No. of Tokens				
NO TDNN		TDNN Variant (FSM =0)		
FSM=2	FSM=4	No. of Blocks=1	No. of Blocks =3	No. of Blocks =5
71.94 / 635466	74.23 / 59164	68.57 / 995153	70.14 / 995153	70.77 / 995153

The TDNN discussed above has not yet been implemented for the real-time system. That is, as present, decisions are based only on the present block, without taking history into account. The best performance for the real-time system appears to be for FSM = 2. In particular, the vowel and the final consonant are generally correctly displayed, whereas the initial consonant is often in error. For the case of FSM=4, the vowel information was not classified correctly and the possible reason for this is also not difficult to find. As mentioned previously, the number of tokens was less than for the cases of FSM=0 and FSM=2 with only one block selected for each component of the CVC including the vowel part. This leads to the case where there are as many consonant blocks as there are vowels. Finally, for the case of FSM=0, only the case where only the current block is considered (No. of Blocks = 1) was tested. The TDNN delay memory has not yet been implemented for the real-time system yet. In this particular case (FSM = 0) recognition of both the initial and final consonants was often in error.

CVC Displays

Two phoneme-based displays have been developed wherein the recognition of the phonemes in the CVC is displayed. They are

1) Whole-Word Display
2) Phoneme Bargraph display.

Whole-Word Display

The whole-word display, primarily intended as a diagnostic display for evaluating the NN training methods, depicts the acoustic waveform and the top 3 phone choices for each segment, one below the other (fig. 11). These phone choices, as explained before, are based on the NN decisions for all the blocks present in the segment. This display is based on a "flow" mode—that is, the waveform and the recognized decisions are represented on the screen as moving from the left to right as the speaker pronounces the CVC utterance over time. This gives the look and feel of "real-time" to the speaker. In reality, there is a delay of a segment (segment time is typically 50ms) before the display is updated with the recognized decision for the blocks present in the particular segment, as mentioned previously in the explanation of the double buffering.

The recognized decisions for each segment are represented as rectangles. The width of the rectangle corresponds to the length of the segment. To determine rectangle heights, the top three-scoring phones (that is the three largest NN outputs) are first normalized so that the sum of the three equals 1.0 Then each of the phones are mapped to a rectangle of height proportional to its normalized value, and with the top scoring phone is represented as the top rectangle, the next highest scoring phone as the middle rectangle, and the lowest scoring phone as the bottom rectangle. The identities of the top 3 choices themselves are listed below the rectangle display.

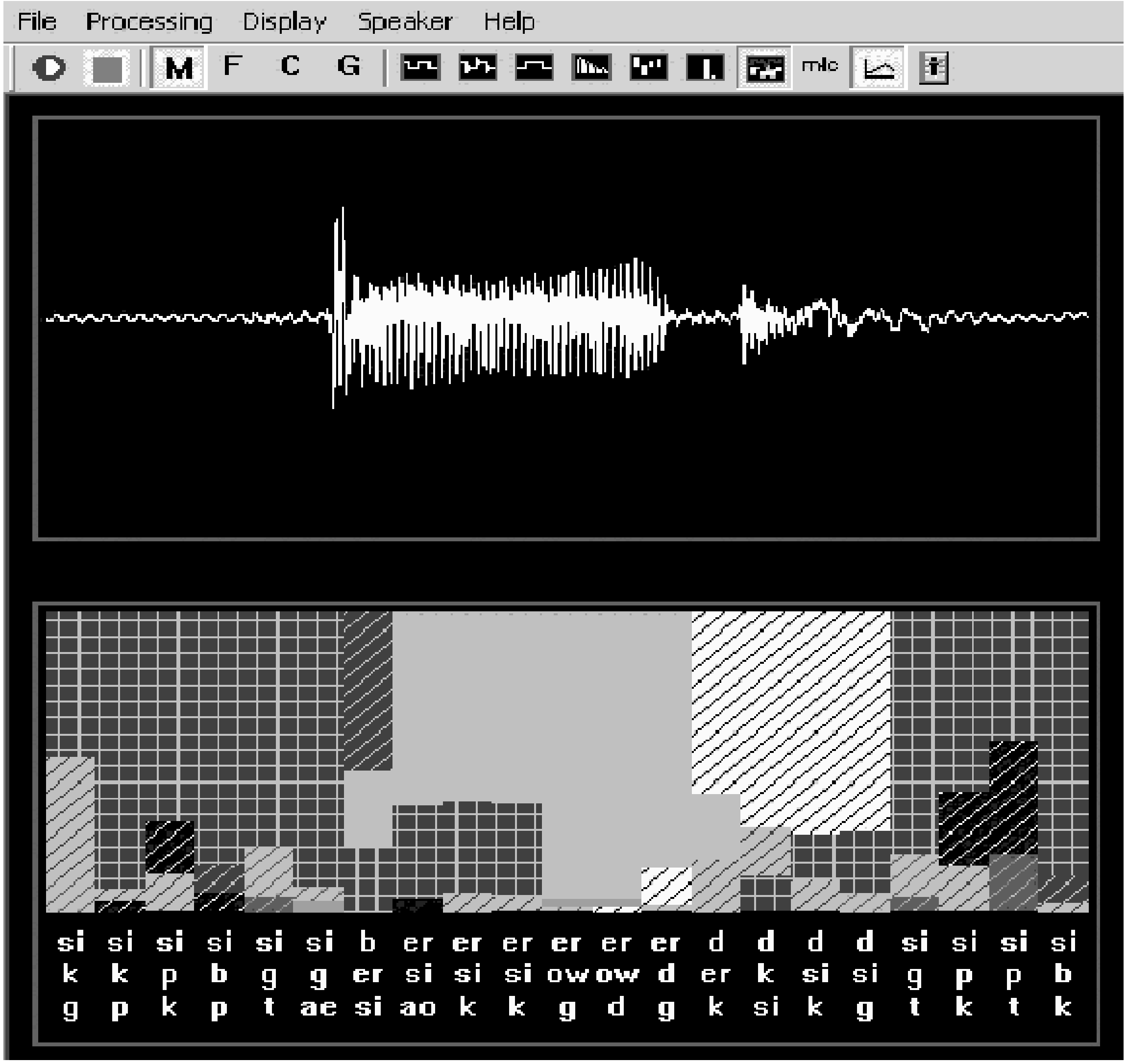
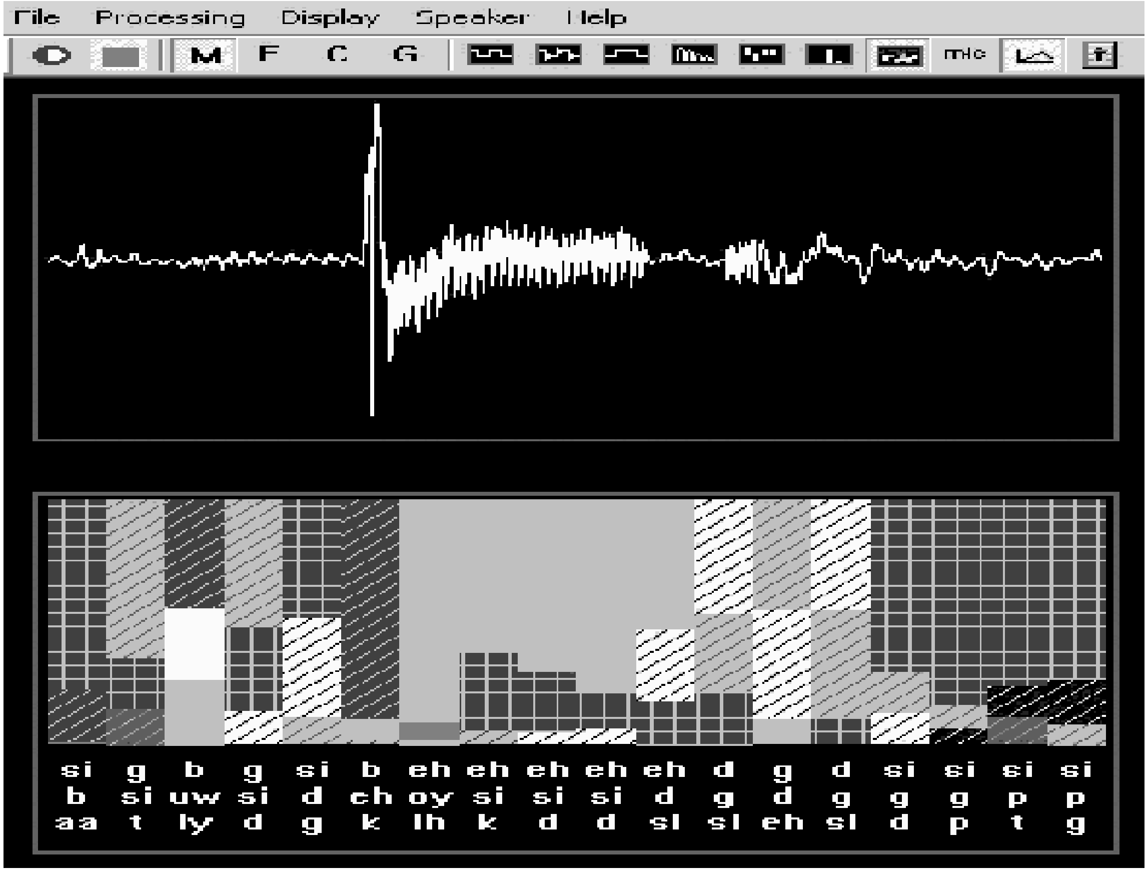

File
Processing
Display
Speaker
Help
M
F
C
G
mlc
si si si si si si b er er er er er er d d d d si si si si
k k p b g g er si si si ow ow d er k si si g p p b
g p k p t ae si ao k k g d g k si k g t k t k

Fig. 11. CVC whole-word display for the word "bird".

The unique sections of the CVC, namely the consonant, vowel and the silence, are distinguished by the texture; while the different consonants and the vowels are distinguished among themselves by their color. This method of distinguishing the different parts of the CVC adds to the clarity as opposed to just using colors to separate each of the 20 phonemes. For example all the vowel segments are displayed as plain colors; all the consonant segments are displayed as different colors interspersed with stripes; silence is represented as horizontal and vertical grid lines crossing each other.

Two figures of this display are shown (figs. 11 and 12) to illustrate the inherent difficulty in recognizing the consonants. Figure 11 shows the pronunciation of the CVC “Bird”. The top choices for all the phonemes present in this word are correct. But the recognized decisions as shown in fig. 12 that represents the word “Bed” are in error with respect to the consonants. This could be attributed to the reason that consonants are very short in duration compared to the vowels and some of the consonants like “b”,”p” sound very similar. The accuracy of stop consonant recognition by automatic algorithms remains a difficult problem for automatic speech recognition software.



si	g	b	g	si	b	eh	eh	eh	eh	eh	d	g	d	si	si	si	si
b	si	uw	si	d	ch	oy	si	si	si	d	g	d	g	g	g	p	p
aa	t	ly	d	g	k	lh	k	d	d	sl	sl	eh	sl	d	p	t	g

Fig. 12. CVC whole-word display for the word “bed”.

Phoneme Bargraph display

The phoneme bargraph display is configured the same as the vowel bargraph display [13], as shown in the following fig. 13. It resembles a histogram with one bar for each of the phonemes. The height of the bar signifies the pronunciation correctness. This display is relatively non-intuitive and cumbersome as compared to the whole-word display largely due to the fact that the consonants are very short in duration, and appear as only brief flashes on the screen. Thus, the user must remember the chronological sequence of the phones.

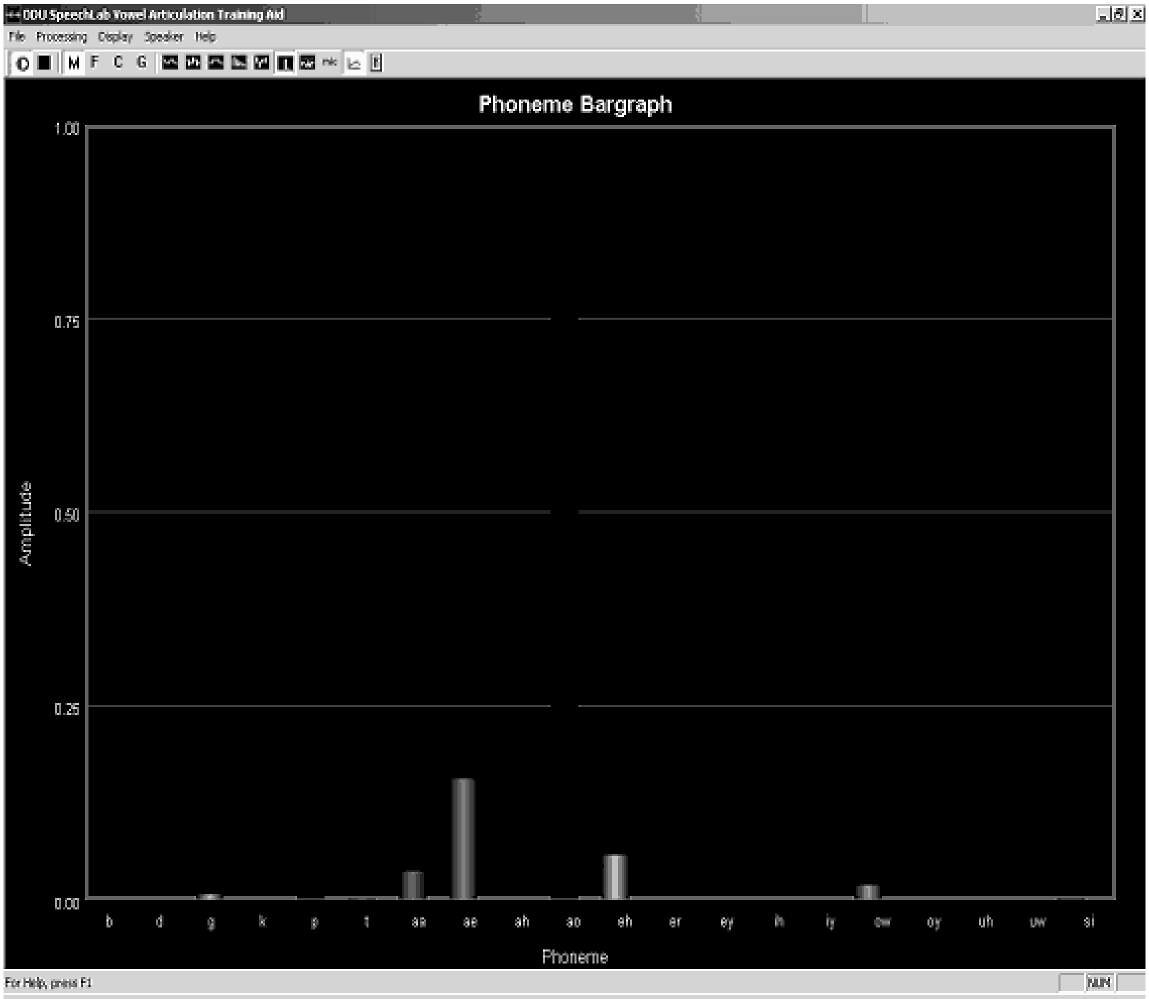


Fig. 13. CVC bargraph display.

Summary

The visual display system giving real-time feedback about the quality of pronunciation of short words (CVCs) was described. The NN training whereby the recognizer based its decision not only on the current block but previous blocks (TDNN) was described and results were compared with the results obtained by using NN similar to the ones used in the VATA. In addition, the CVC display processing steps, signal processing and examples of the displays were shown. At present, the real-time display does not make use of the past history, as the TDNN is capable of.

CHAPTER V

CONCLUSIONS

The improvements brought about in VATA and the attempt made to extend VATA to handle short words has been described. This extension of VATA was brought about as a result of the non-naturalness and clear limited scope of vowels produced in isolation. The training methodologies and the step-by-step procedure for HMM-based modeling for computer-based articulation and the steps involved in building a real-time display system were described in detail. Experiments conducted to improve the VATA and the initial results obtained with the CVC display system were reported. The following were the topics of those experiments:

- Effect of pruning the vowel database by getting rid of mispronounced and noisy speech data on the training as well as the operation of the real-time system.
- Spectral instability in VATA that resulted in jitter in the spectrum.
- An attempt at finding a improved feature set using spectral moments instead of DCTCs.
- Effect of using the identically sampled tokens (11khz) for NN training by downsampling the higher sampled ones (22khz).
- To compare the heuristic labeling of the CVC database to that of using HMM-based triphone models to fix the labels of the phones present in the CVC database collection.
- Find the effect of using a variant of TDNN on the phoneme training.

These experiments were conducted and the results obtained were reported in the previous chapters of this thesis. Based on those results the following conclusions may be drawn:

- Bad data removal greatly improved the training results as well as the operation of the real-time system.
- Spectral stability was improved by using an adaptive length time-smoothing algorithm and this greatly decreased the jitter.
- Spectral moments as a feature alternative to DCTCs was explored but was dropped on the basis that the recognition results were very low and that closer examination revealed a similarity to the DCTCs.

- Downsampling of the higher sampled tokens did result in improved recognition in training as well as in the operation of the real-time system.
- The step-by-step HMM-based triphone modeling procedure improved the labeling of the phones in the CVC database over the heuristic algorithm and was confirmed with the recognition results.
- Employing the TDNN variant did not result in an improvement in the training results but the results indicated that improved recognition was obtained as the time window (number of blocks) is increased.

The real-time CVC display system is at its nascent stage and there is a lot of scope for improvement. A few of them are listed below:

- Inclusion of the TDNN variant in the real-time system and investigate its usefulness to the phoneme recognition particularly with respect to the consonants.
- Since the training recognition percentages are low, the CVC database training set need to be increased.
- A combination of HMM and NN classifier could be used; this needs to be investigated in detail to determine if it can improve phoneme recognition.
- A few more displays need to be built, particularly game displays that interest children

REFERENCES

[1] C.V. Hudgins, “Visual Aids in the correction of speech,” The Volta Review, 1935, Vol. 37, pp. 637-704.

[2] D.J. Povel and N.Arends, “The Visual Speech Apparatus: Theoretical and practical aspects,” Speech Communication, 1991, Vol. 10, pp. 59-80.

[3] R.S. Nickerson and K.N. Stevens, “Teaching speech to the deaf: Can a computer help?,” IEEE Transactions on Audio and Electroacoustics, 1973 ,AU-21, pp. 445-455.

[4] R.S. Nickerson, D. N. Kalikow and K. N. Stevens, "Computer-aided speech training for the deaf," Journal of Speech and Hearing Disorders, 1976, 61, pp. l20-l32.

[5] R.S. Nickerson, K.N. Stevens, and A.M. Rollins, “The BBN computer-based system of speech training aids for the deaf: Current uses,” Research Conference on Speech Processing Aids for the Deaf, 1977 (Washington, D.C, May).

[6] B.Maassen, and D.J. Povel, “The effect of segmental and suprasegmental corrections on intelligibility of deaf speech,” Journal of the Acoustical Society of America, 1985, Vol. 78,pp. 877-886.

[7] D.J.Povel, and M.Wansink, “A computer-controlled vowel corrector for the hearing-impaired,” Journal of Speech and Hearing Research, 1986,Vol. 29, pp. 99-105.

[8] D.J.Povel, “Development of a vowel corrector for the deaf”, Psychological Research, 1974a, Vol. 37, pp. 51-70.

[9] D. Kewley-Port, C.S. Watson, M. Elbert, K. Maki and D. Reed, “The Indiana Speech Training Aid (ISTRA) II: Training curriculum and selected case studies,” Clinical Linguistics and Phonetics, 1991,5, pp. 13-38.

[10] A.M.Zimmer, S.A.Zahorian and B. Dai, “Personal computer software vowel training aid for the hearing impaired, “ International Conference on Acoustics, Speech, and Signal Processing, 1998, Vol. 6, pp. 3625-3628.

[11] G.E. Peterson and H.L. Barney, “Control methods used in the study of the vowels,” Journal of the Acoustical Society of America, 1952, Vol. 24, pp. 175-184.

[12] S.A. Zahorian and Z.B. Nossair, “ A Partitioned neural network approach for vowel classification using smoothed time/frequency features,” IEEE Transactions on Speech and Audio Processing, 1999,Vol. 7, No. 4, pp. 414-425.

[13] A.M. Zimmer, “VATA: An improved personal computer-based vowel articulation training aid,” Unpublished Masters Thesis, Old Dominion University, 2002, Norfolk, VA.

[14] A. Papoulis and S.U. Pillai, “Probability, Random Variables and Stochastic Processes,” 4th edition, 2002, McGraw-Hill, New York.

[15] M. Devarajan, F. Meng, P.Hix and S.A. Zahorian, “HMM-Neural network monophone models for computer-based articulation training aid for the hearing impaired,” International Conference on Acoustics, Speech, and Signal Processing, 2003.

[16] S. Evangelos, N. Fakotakis and G. Kokkinakis, “Fast endpoint detection algorithm for isolated word recognition in office environment, International Conference on Acoustics, Speech, and Signal Processing, 1991, pp. 733-736.

[17] K. Kasi and S.A. Zahorian, “Yet another algorithm for pitch tracking,” Internation Conference on Acoustics, Speech and Signal Processing, 2002.

[18] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev and P. Woodland “ The HTK Book (for HTK version 3.1),” Cambridge University Engineering Department, 2001.

[19] S.A. Zahorian and A. Jagharghi, “Spectral-shape features versus formants as acoustic correlated for vowels,” Journal of the Acoustical Society of America, 1993, 94-4, pp. 1966-1982.

VITA

MUKUND DEVARAJAN

DEGREES:

Bachelor of Science (Electronics and Communication Engineering), Madras University, Madras, Tamilnadu, India, May 1998

PROFESSIONAL CHRONOLOGY:

Department of Electrical and Computer Engineering
Old Dominion University, Norfolk, Virginia
Graduate Research Assistant, January 2002 – Present
Graduate Teaching Assistant, January 2002 – May 2003

Iflex Solutions Ltd., A Citicorp venture capital company
Bangalore, Karnataka, India
Associate Consultant, March 1999 – July 2001

Spark Solutions Ltd.,
Madras, Tamilnadu, India
Associate Consultant, May 1998 – February 1999

SCHOLARLY ACTIVITIES COMPLETED

Devarajan, M., Meng, F., Hix, P., and Zahorian, S.A. (2003). "HMM-Neural network monophone models for computer-based articulation training aid for the hearing impaired," International Conference on Acoustics, Speech, and Signal Processing (Also accepted to the ICME conference 2003).