CANADIAN THESES

THÈSES CANADIENNES

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30.

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30.

THIS DISSERTATION HAS BEEN MICROFILMED EXACTLY AS RECEIVED

LA THÈSE A ÉTÉ MICROFILMÉE TELLE QUE NOUS L'AVONS REÇUE

Morphological Representation of Speech Knowledge for Automatic Speech Recognition Systems

Mathew Joseph Palakal

A Thesis

In

The Department

of

Computer Science

Presented in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy at
Concordia University
Montréal, Québec, Canada

May 1987

ISBN 0-315-37062-9

ABSTRACT

Morphological Representation of Speech Knowledge for Automatic Speech Recognition Systems.

Mathew Joseph Palakal, Ph.D.
Concordia University, 1987

Speaker-independent recognition of large or difficult vocabularies by computers is still an unsolved task, even if the words are pronounced in an isolated manner. Using existing knowledge about production and perception of speech, phonemes, diphones and syllables can be useful for conceiving prototypes of Speech Units. Speech Unit prototypes can be characterized by a redundant set of Acoustic properties.

Automatic Speech Recognition (ASR) systems based on acoustic property descriptors is not very efficient if the set of properties used and the algorithms for their extraction are not well chosen and conceived. For this reason, it is worth investigating property descriptors based on those properties that are expected to be robust speaker-independent cues of fundamental phonetic events.

Speech spectrogram is an invaluable tool in ASR research and it contains rich acoustic and phonetic knowledge about speech. Expert human spectrogram readers are able to interpret speech spectrograms by visual examination. The interpretation is usually based on the experts linguistic knowledge and correlating this knowledge with the characteristic pattern of speech. Machines can have similiar capability if patterns of various speech units can be collected, described, and learned statistically.

Based on the above considerations, this work proposes a paradigm for the extraction and interpretation of speech knowledge contained in speech spectrograms. The model proposed attempts to integrate knowledge-based extraction of relevant speech properties and statistical modelling of their distortions.

The speech spectrograms are considered as patterns and knowledge contained in these patterns is described as morphologies and represented as a taxonomy. The recognition model uses a frame-work of Procedural Network which uses networks of actions performing variable depth analysis and integrates cognitive and information-theoretic approaches. Experimental results are reported for a large number of speakers using digits and letters as test data.

ACKNOWLEDGEMENTS

I wish to express my most sincere gratitude to my thesis supervisor, Prof. Renato DeMori for his superior guidance and insight throughout the span of this research. His commitment and dedication to research has given me the motivation and encouragement to successfully complete this endeavor. I wish to thank him for all the financial and moral support without which I could not have carried out my studies. Above all I would like to thank him for providing a harmonious working relationship. His patience and understanding, both academic and personal, have been most valuable for completing this work within a reasonable amount of time.

I would also like to thank the staff of Centre de Recherche Informatique de Montreal Inc., especially, Lynn-Marie Holland, Kathy Cameron, Nadine Lasalle, Louise Doyon, Bernard Turcotte, Michel Savoie, Marc Comeau and Mario Vachon for all their help and kindness.

My sincere thanks to Betty Hacket and Allen Kowalski for reading this manuscript and providing me with valuable suggestions.

During my long period of studies at Concordia University, I received much help from Stephanie, Halina, Mary, Terry, and Angie, the secretaries of the Computer Science department. I wish to thank them most sincerely.

My fellow students and colleagues, Ettore Merlo and Jean Rouat, were of great assistance in clearing up many ideas through several valuable discussions.

I would like to thank all my friends in Montreal, who lent their support morally and otherwise and helped make this thesis a reality.

Last but not least, I thank my wife Philomina for spending many days on the Macintosh helping me to prepare this thesis.

DEDICATED TO MY
LOVING FAMILY
AND
CARING FRIENDS

TABLE OF CONTENTS

List of Figures

List of Tables

Chapter 1

Introduction

The ultimate goal of research on Automatic Speech Recognition (ASR) is to give the machine similar capabilities of humans to communicate in natural spoken languages. Such research is of great interest because of two main reasons: the application point of view and research point of view.

Since speech is our most natural mode of communication, we should have the potential of machines that more fully accomodate to the human user, rather than perpetuating the trend of our mechanical slaves actually enslaving us in unwanted diversions, such as learning keypunching, typewriting, and complex programming methods [1]. As an application point of view, there are several advantages of voice input to machines: voice input allows eyes and hands free, needs little or no user training, permits fast, multimodal communication, freedom of movement and communication.

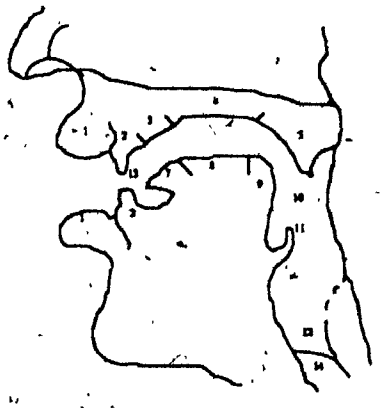
Speech recognizing machines have possible application in many areas, like, office automation, assembly line inspection, airline reservation and aids for the handicapped.

From research point of view, automatic speech recognition is a difficult problem extending over the past 4 decades. Even though significant progress was made in the past, the ultimate goal, a perfect listening machine, is yet to be achieved. Several areas of human perception of voice have yet to be explored and the findings of such research must be exploited for building listening machines. What has been done so far is mostly based on analytical methods and only very recently researchers have incorporated detailed speech knowledge in their recognition models.

1.1 Speech and Speech Knowledge

Fig. 1.1 shows the organs of our speech production system. Speech sound is produced when air flows through and resonates in the vocal tract. Different sounds are

produced because of different vocal tract configurations. For a class of speech sounds, like vowels, there is a set of resonant frequencies characterizing each sound in the class. Also, different sounds are produced depending upon the source of excitation. During speech production, the articulators move continuously, rather than discretely, resulting in a continuum of acoustic characteristics.



1. Lips
2. Teeth
3. Teeth-ridge
4. Hard plate
5. Velum
6. Uvula
7. Blade of tongue
8. Front of tongue
9. Back of tongue
10. Pharynx
11. Epiglottis
12. Vocal cords
13. Tip of tongue
14. Glottis

Fig. 1.1 Organs of Speech production

There are two basic types of sound sources in speech: periodic vibration of vocal folds and turbulent noise. When the speaker exhales, air passes through the larynx. If the glottis is partially closed, then the air passing through the constriction cause the vocal cords to open and close quasi-periodically, producing the voiced sound. The rate of vibration, which is controlled by vocal-cord tension, is called the fundamental frequency or pitch.

When excitation is at the glottis, the vocal folds remain open and causes a weak turbulence, to produce aspiration sounds. A constriction in the vocal tract causes turbulent noise, which has a flat spectrum, and is called a frication sound.

The peaks in the spectrum of voiced sound are called formants and are labelled as F_1, F_2, ...F_i, where F_1 is called the first formant and so on. The lips, tongue, jaw, and velum can be moved to change the shape of the vocal tract. The resultant vocal tract acts

as a cascade of resonators, which filter the source. The poles of the vocal tract transfer function generate spectral peaks called the formants. In the case of nasals, sound passes through the nasal cavity, but the mouth cavity, which is closed, acts as a side branch and introduces zeros in the spectrum. The interaction of the poles and zeros can change the frequency of the formants [2].

Early speech scientists described speech sounds in terms of particular characteristics of speech-like, voiced, unvoiced, front, back, etc. [3], [4].

During speech production, the articulators move relatively slowly from one position to another. The articulators often do not reach their "target" positions due to contextual effect of neighboring phones: this is called coarticulation [5]. Therefore, the spectral sequence associated with a particular phone can vary widely depending on the adjacent phones.

Different speaker's vocal apparatus can vary in terms of the source spectrum, the length of the vocal tract, and the relative shape of the vocal tract. For this reason, the speech of adult males and females differ, typically, the pitch period of female speech is about 20% shorter causing an average 20% increase in the formant frequencies [6].

In addition to the differences due to speaker, dialect, and phonetic context, there is also random variation in the pronounciation of speech sounds. Even for the speech of a single speaker, the spectral properties present cannot be converted back to a phonetic string without the use of higher-level knowledge.

The articulatory movements vary for different speech sounds: for some the vocal tract configurations are stable, while for others, they are not. For example, the configuration for sound /m/ is more stable than for an /r/. In vowel to nasal transition for /m/, the velum is lowered while sound /r/ is produced by retroflexing the tongue. The tongue cannot move as fast as velum and this causes the difference in the configuration dynamics.

1.2 Acoustic Characteristics of Phonemes

The phonemic classification chart in English is shown in Fig 1.2. A brief discussion about the acoustic nature of each phonetic group is considered now. Most of the following information is summarized from [2], [7], [8], and [9].

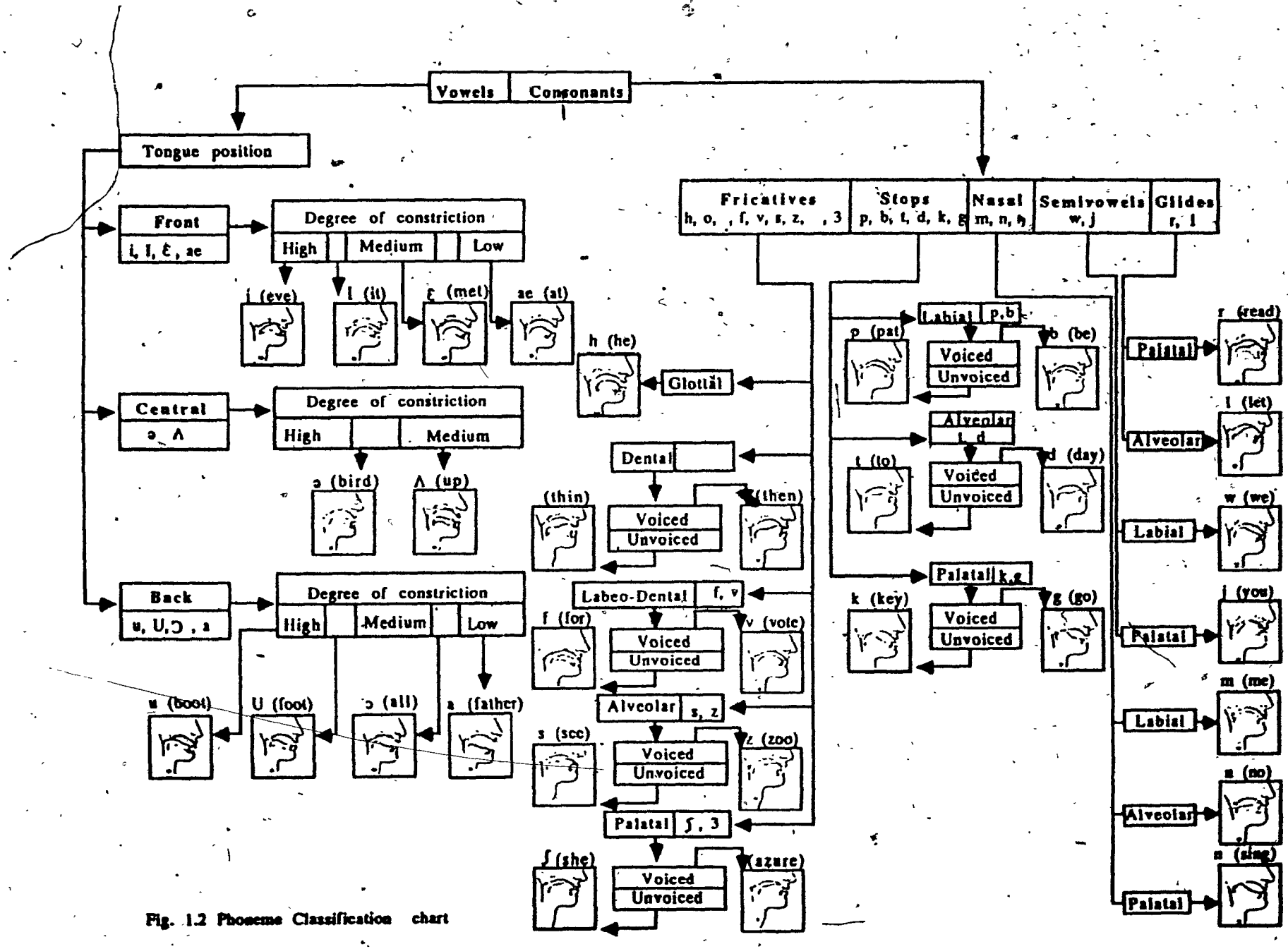


Fig. 1.2 Phoneme Classification chart

Vowels

Vowels are produced by exciting the open vocal tract with a periodic (voiced) source. Vowels are often characterized by substantial energy in the low- and mid-frequency regions. The energy in the high frequencies above 3500Hz are less important for vowel characterization. The characteristics of different vowels depend on the location of the tongue, position of the jaw, and the degree of lip rounding. The resulting shape of the vocal tract determines the formant frequencies. The three classes of vowels, back, central, and front, occur as a result of the tongue position: in general when the tongue moves forward, the second formant rises; as the tongue moves higher or the jaw rises, the first formant decreases. Lip rounding lowers all of the first three formants.

Fig 1.3 shows acoustic waveforms and Fig 1.4 shows the resonance characteristics of vowels [8].

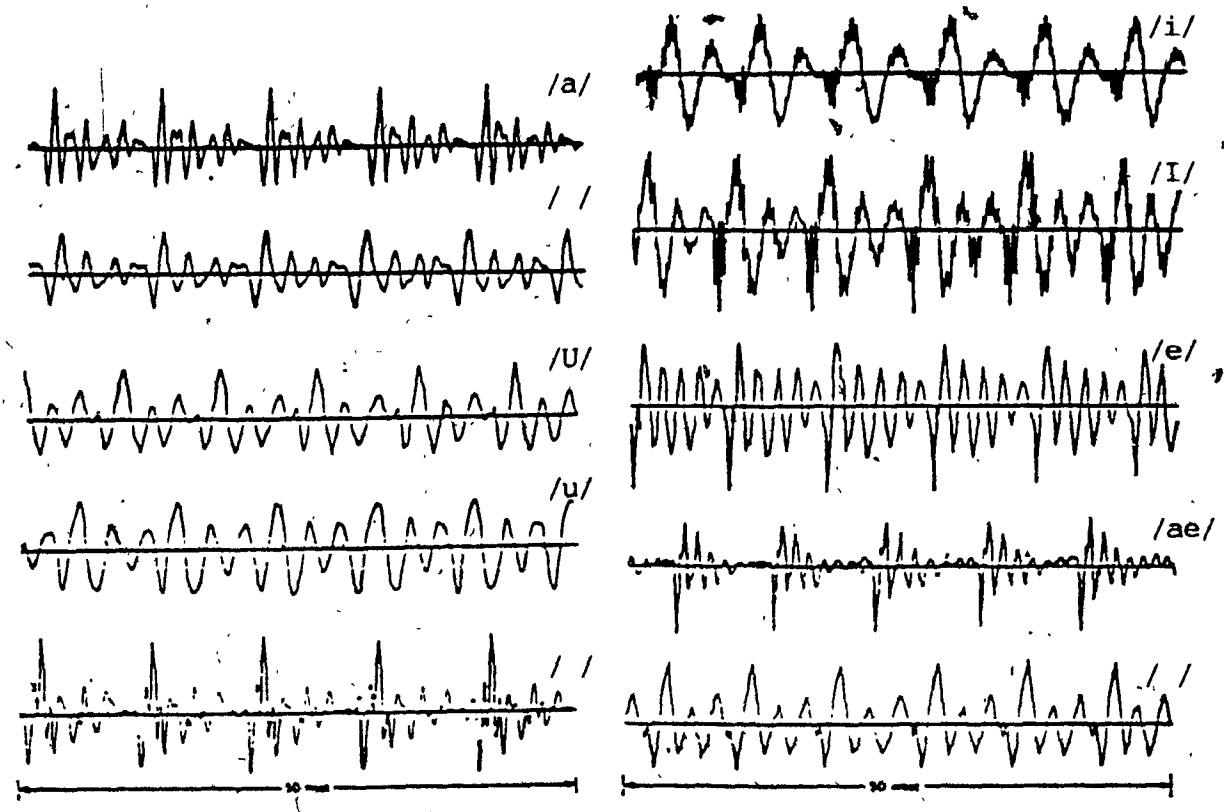


Fig 1.3 Acoustic waveforms of vowels

Many vowel recognition methods measure the first three formants in the middle portion of the vowel and compare those values against stored targets. These values are

called vowel loci. Variances of formant frequency distributions for each vowel around vowel loci are speaker independent.

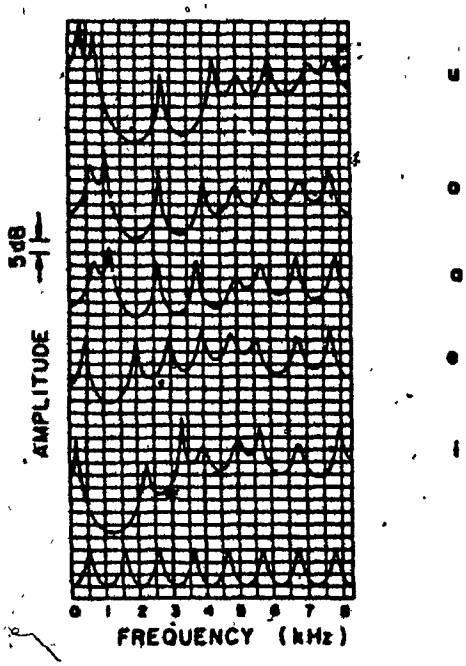


Fig 1.4 Resonance characteristics of vowels

Consonants

The consonants are divided into several groups depending on the manner in which they are articulated. The five such groups in English are, plosives, fricatives, nasals, glides, and affricates. Consonants from different "manner-of-articulation" groups often have different acoustic correlates. Consonants within a "manner-of-articulation" group differ in their voicing characteristics and the position of constriction. The acoustic properties of consonants differ both within the consonants and in the adjacent vowels in the form of formant transitions. This problem must be considered in order to recognize consonants.

Nasals

The nasals (m, n, ng) are always adjacent to a vowel, and are marked by a sharp change in intensity and spectrum, corresponding to the closing of the oral cavity and opening of the velum. Nasal sound is produced by a glottal excitation and vocal tract

constriction at some point. By lowering the velum, the air flow is forced through the nasal cavity.

Nasals are very difficult to recognize since nasal murmur differs significantly from speaker-to-speaker, because of differences in the size and shape of nasal and sinus cavities. Nasal murmur is also heavily affected by phonetic environment.

Some of the nasal characteristics are, the prominence of a low-frequency spectral peak at around 300 Hz, little energy present above 3k Hz, sharp spectral discontinuity between the nasal murmer and the adjacent vowel. Fig 1.5 shows acoustic waveforms of nasal sounds /m/, /n/.

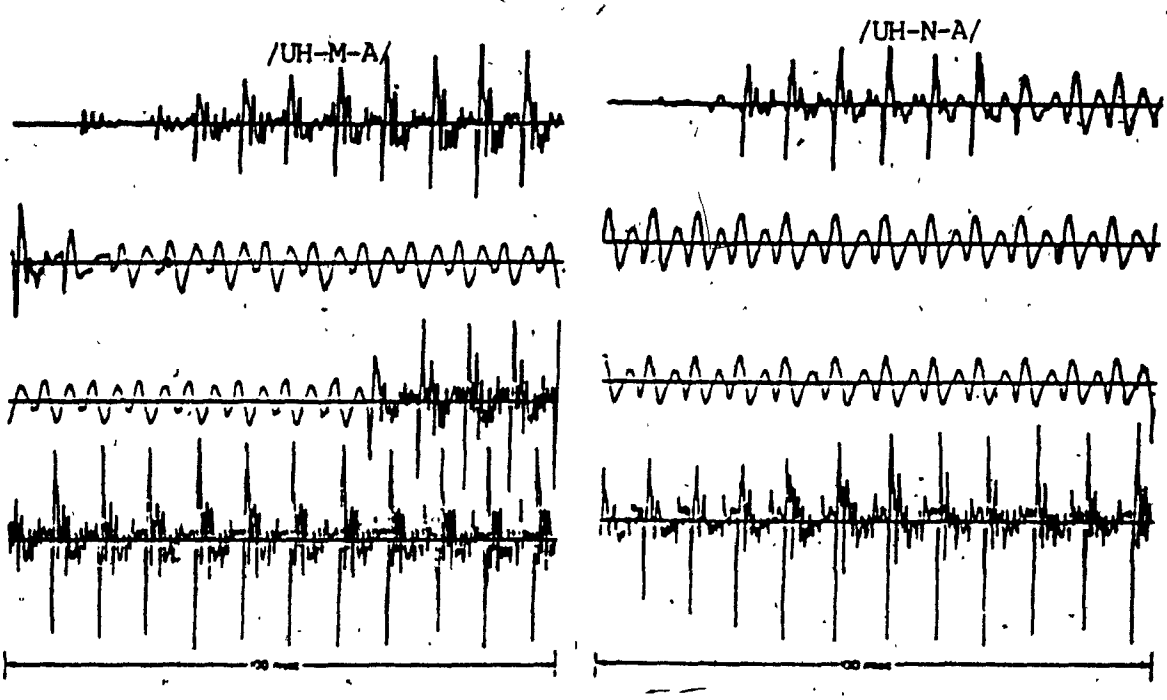


Fig. 1.5 Acoustic waveforms of nasal sounds

Liquids and Glides

This group is also sometimes known as semi-vowels. Such consonants often appear next to vowels as in the case of nasals. These sounds are produced by a constriction in the vocal tract which is smaller than that of vowels but still large enough so that no turbulance is generated. Each phoneme in this group has a close association with certain vowels such as,

/w/ ⇒ /u/, /y/ ⇒ /i/

/r/ ⇒ /ɜ/, /l/ ⇒ /o/

The distinguishing characteristics of these consonants from other consonant groups are: the rate of articulatory movement is considerably slow which implies slower formant transitions; the formants of these sounds have the following qualitative relation with respect to the formants of adjacent vowels.

- /w/ has lower F_1 and F_2
- /l/ has low F_1 and F_2 with higher F_3
- /r/ has lower F_3
- /y/ has very low F_1 with higher F_2.

The formant patterns within the phonemes are similar to some vowels and their distinguishing characteristics are often detected by comparison with those of the adjacent vowels.

Plosives

Plosives are also known as the "stop" consonants. Plosive sounds are classified into two groups: voiced or lax (/b/, /d/, /g/) and unvoiced or tense (/p/, /t/, /k/). Voiced Plosive sounds are produced by building up pressure behind a total constriction somewhere in the oral tract. During the total constriction no sound is radiated through the lips. However, a small amount of low frequency energy is radiated through the walls of the throat called the voice-bar.

Unvoiced plosive sounds are produced in the same way as voiced sounds, except that during total constriction, vocal cords do not vibrate. Plosive consonants are considered to be the most difficult consonants to recognize because of the following reasons:

- the production of stop is dynamic involving a closure and release period.
- the complicated nature of this production results in many diverse acoustic cues.
- the acoustic events during the production of the sound can be omitted or

severely distorted.

Some of the characteristics of voiced and voiceless stops are:

a. the plosives are characterized acoustically by a period of prolonged silence, followed by an abrupt increase in amplitude at the consonantal release. The release is accompanied by a burst of frication.
b. for voiceless stops, the aspiration noise is generated at the glottis.
c. the voice-onset-time (VOT), which is the duration between the release and the onset of normal voicing for the following vowel is longer for unvoiced (30 to 60ms) than for voiced (10 to 30ms).
d. the voiced stops are often prevoiced creating the voice bar in the low-frequency region.
e. the amplitude of the burst is significantly different between voiced and voiceless plosive sounds.

Fig 1.6 shows acoustic signal for samples of voiced and voiceless plosives.

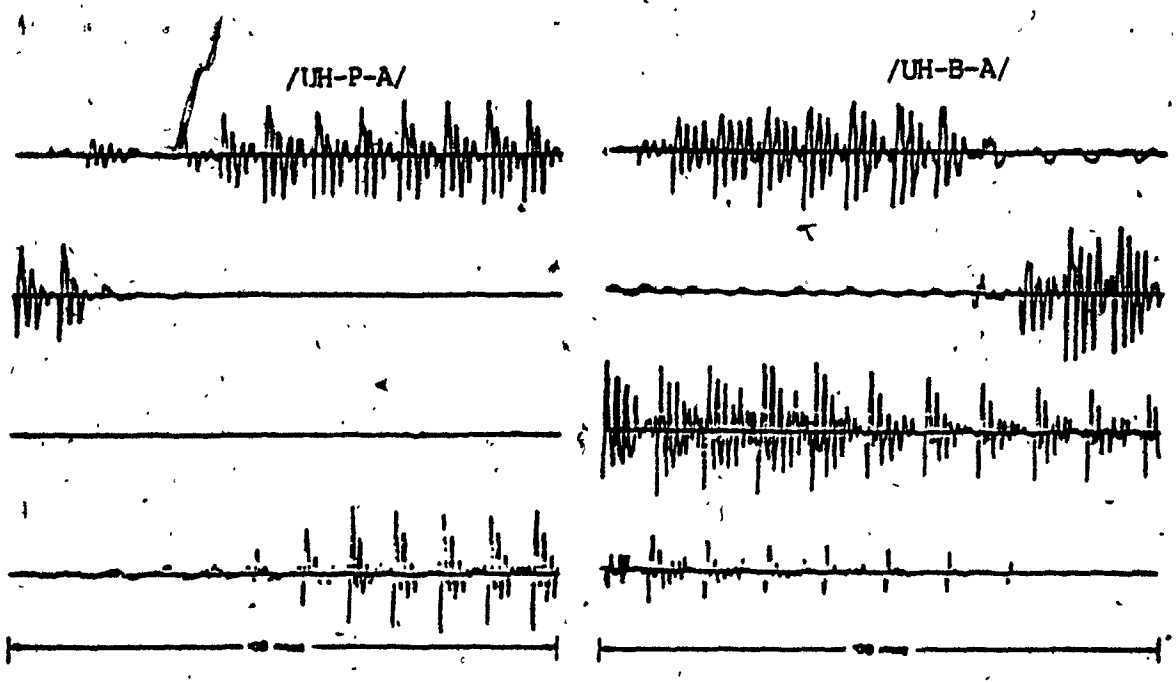


Fig. 1.6 Acoustic waveforms of plosive sounds

Fricatives

Like plosives, there is a voiced fricative group (/v/, /TH/, /z/, /zh/) and a voiceless fricative group consisting of (/f/, /o/, /s/, /sh/).

Unvoiced fricatives are produced by exciting the vocal tract by a steady air flow which becomes turbulent in the region of a constriction in the vocal tract. Unlike the unvoiced fricatives, voiced fricatives are produced by vocal cord vibration and excitation at glottis. Since vocal tract is constricted at some point, air flow becomes turbulent.

Voiced fricatives often have simultaneous noise and periodic excitations which causes great amount of low-frequency energy in the beginning of frication. Voiced fricatives are also shorter than unvoiced fricatives.

Acoustic signal for some of the fricative sound are shown in Fig 1.7.

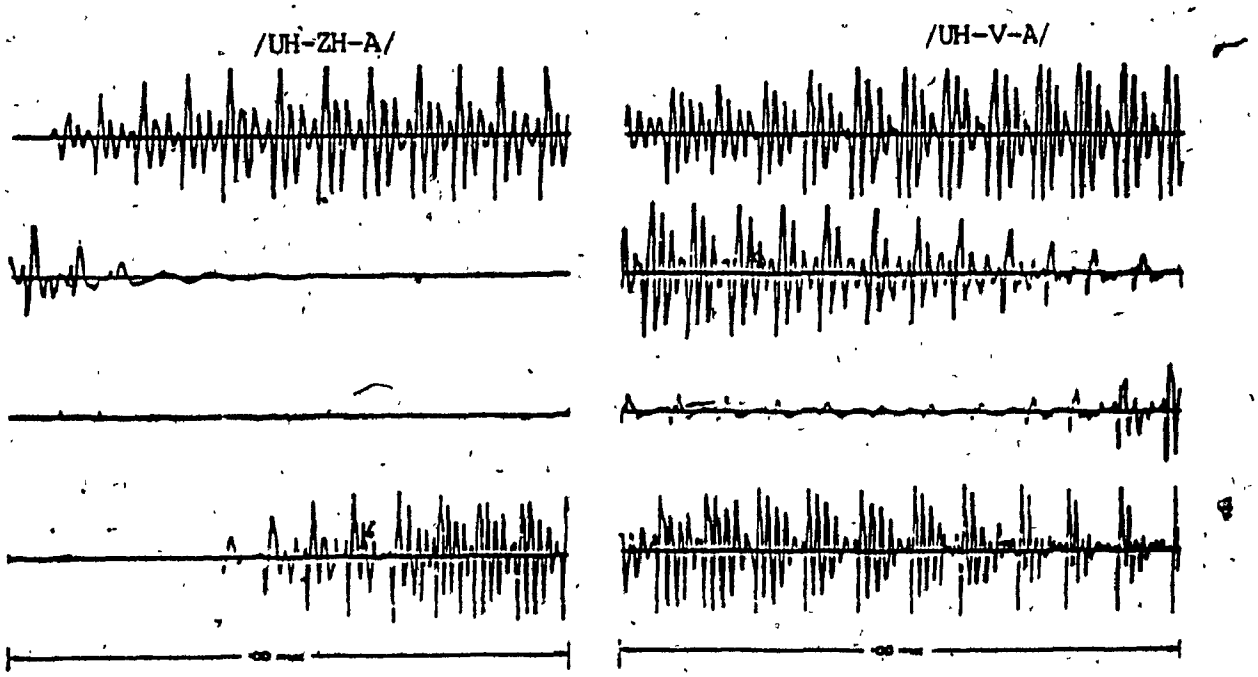


Fig. 1.7 Acoustic waveforms of fricative sounds

Affricates

The affricates (/c/, /j/) are often considered as a plosive followed by a fricative.

These sounds are often modeled as a sequence of two phonemes (/c/ as /t-s/ and /j/ as /d-z/).The duration of frication is often very short as compared to other fricatives.

The properties of each class of phonemes can be considered as acoustic knowledge about the speech sounds. However, these properties are not independent and they may be distorted or omitted. Because the cues for a phoneme are so redundant, the human speaker tends to be rather careless about producing these prototypical features for a given phoneme. Distortions may vary from speaker to speaker, or even over time for the same speaker.

Despite the above mentioned problems regarding human speakers, human listener somehow has no trouble discarding the "bad" features and accepting only the "good" ones. This is possible because "higher level context" is available, and also humans use phonotatic constraints in decoding distorted syllables. This is a clear indication that, enough information necessary to decode the phonemes is present in the acoustic signal.

Therefore, phonetic recognition algorithms must consider several features jointly, rather than "a particular feature". Given several features that each contribute towards making phonetic distinctions, the Acoustic Phonetic Recognizer must enlist the aid of a multidimensional feature selection and pattern recognition algorithm to design the optimum classifier [2].

Many approaches have been used to incorporate various features that are present in speech sound. Some of those important techniques are discussed in the next section.

1.3 Speech Recognition Systems

During the past two decades, there has been substantial progress toward the goal of constructing machines capable of understanding/recognizing human speech. One of the key improvements has been the development and application of mathematical methods that permit modeling the speech signal as a complex code with several coexisting levels of structure.

For any speech recognition system, the spectrum is usually represented by Fourier coefficients, zero crossing rate, or the parameters of some local model of signal such as linear prediction coefficients. Temporal information can be directly obtained as in the case of voice onset time. Prosodic information is often extracted by estimating fundamental frequency to represent pitch and the logarithm of energy integrated over 45-ms intervals to measure intensity [10].

Presently, features obtained this way are neither robust nor are they invariant

with respect to speaker. As a result of some psychophysical experiments, [11], there is an assertion that speech is a composite signal, hierarchically organized so that simpler patterns at one level are combined in a well-defined manner to form more complex patterns at the succeeding level. Such an organization strategy is easily explained in terms of information theoretic principles.

The structures at each level of the hierarchy serve to constrain the ways in which the individual patterns associated with that level can be combined. The constraints build redundancy into code, thereby making it robust to errors or variations caused by a speaker. This way relatively few primitive patterns can be combined in a multilevel hierarchy according to a complex code to form a rich, robust information-bearing structure [10].

Spectra and prosodics, the primitive patterns according to linguistic theories, can be combined in several ways to form phonemes [12], [13], [14], broad phonetic catagories [15], [16], diphones [17], demisyllables [18], [19], syllables [20], supra-segmental phrases [21], [22], and sentences [23], [24], [25]. For the implementation of theories, data structures such as templates [26], [27], [28], formal graminars [24], [29], [30], Markov chains [31], [32], [33], fuzzy sets [35], and hash tables [36] have been used.

Parametric and Non-parametric Methods

In nonparametric methods the primitive measurements of the speech signal can be compared without regard for their temporal location. However, sequences of these measurements, such as the one required to represent speech signals of greater temporal extent must, due to the nonstationarity of the speech signals, take account of time to be meaningfully compared [10].

A comparison approach between the two methods, according to Levinson [10] is shown in Fig 1.8.

It has been argued that, nonparametric methods are easy to train, but as a classifier the parametric methods perform just the opposite in terms of complexity.

Template Matching, Stochastic Modeling, and Probabilitic Parsing were the most successful models. Some of the benchmark systems developed using the above approach are described briefly in the following sections. Most of the following discussions are summarized from [37], [38], [39].

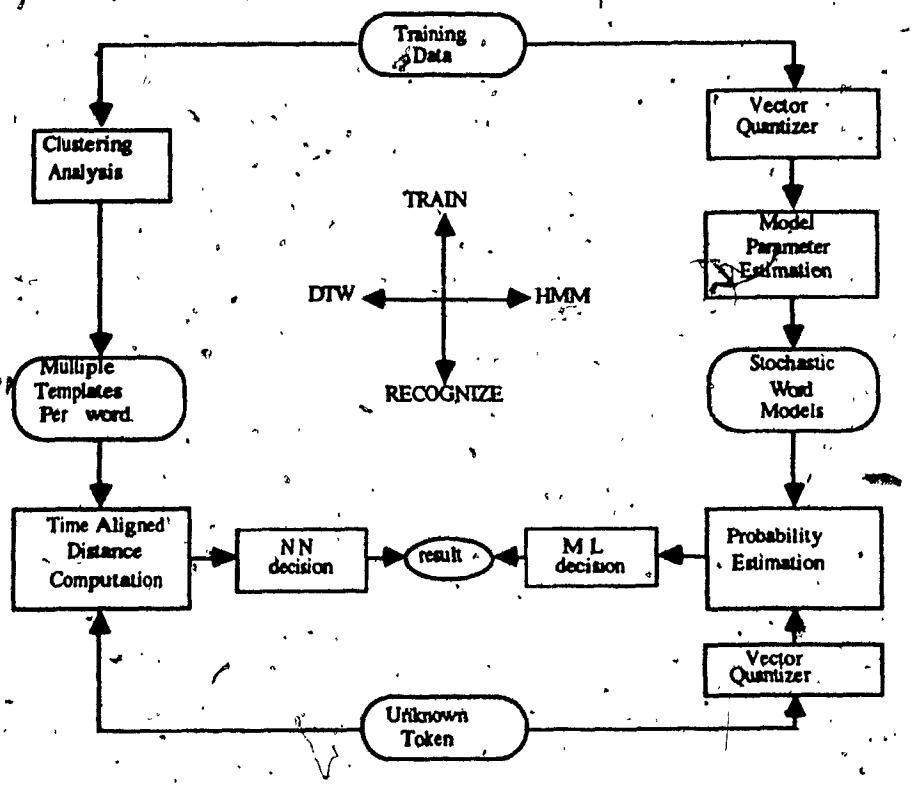


Fig. 1.8 Comparison of Parametric and Nonparametric methods
(After S.E. Levinson [30])

1.3.1 Template Matching

Template matching is based upon principles of nonparametric estimation of likelihoods by means of invariant metrics. In template matching, each recognition unit (a unit could be a word or a phoneme etc.) is represented by at least one template, created from a set of training utterances. Each template contains a sequence of patterns extracted in time. The patterns are spectral and/or prosodic features. During matching processes, the target pattern is matched against stored templates, and the matched template with a minimum distance is the selected candidate. The whole success of the pattern matching approach lies in the comparison process.

Absolute Pattern Match

The most basic comparison process is simply to correlate the time-frequency word patterns produced by a pre-processor in order to determine the distance between an unknown word and each template. This may not be possible because words are often of

different duration and their corresponding patterns are of different sizes. A potential solution to this problem can be obtained by aligning the beginnings of all the patterns, and by correlating only over the areas of overlap. This simple technique, experimented by White and Fong [40], requires *N* vector comparisons per pattern match, where *N* is the number of vectors in the smallest pattern.

Best Absolute Time Alignment

An alternative to aligning the beginnings of words in order to perform an absolute comparison, is to adjust their relative timing to maximize the correlation of the overlap. That is, starting with the beginnings aligned, the patterns are shifted with respect to each other until the ends align. The similarity of the pattern overlap is calculated at each shift, and the highest similarity is the result of the comparison. Computationally this scheme is expensive. Recognition experiments carried out using this method by Moore [37] found no significant improvements in recognition.

Linear Time-Normalization

The previous techniques do not consider the fact that the same word is very rarely the same duration on different occasions. In order to handle this problem, each pattern is uniformly time-normalized to make them the same size. This is known as linear time-normalization.

For practical applications, either the template patterns are time-normalized to the unknown pattern, or all patterns are time-normalized to a pre-set duration.

If a very small vocabulary (10 to 30 words) is used, such techniques perform well. A commercial system made available by Interstate Electronics called VRM used linear time-normalization approach. Several other commercial systems used such techniques. The performance also depends on the inherent confusability of the words, consistency of speakers, type of features used, and the number of training samples allowed.

Non-linear time-Normalization

Linear time-normalization do not perform well for larger vocabularies. The reason is that making the pattern of fixed length is not an adequate model of what actually happens when people make words longer or shorter. A model of time scale

distortion which allows different sounds to be distorted differentially would align the pattern more meaningfully.

One approach to computer recognition of speech requires that we compare two sequences of elements and compute the distance between them by finding an optimal alignment or correspondence between the elements of one sequence and those of the other. In speech research, these sequence comparison methods are capable of performing nonuniform *time warping* using *dynamic programming* (DTW). The name refers to allowing nonlinear distortions of time scales in computing the acoustic similarity between a reference prototype and features extracted from the input utterance, thereby taking into account speaking rate variability as well as substitution, insertion, and deletion errors: dynamic programming is an efficient algorithm by which both optimal alignments and the resulting distances are computed at the same time. Data and prototypes to be matched are represented by discrete sequences produced after either synchronous or asynchronous sampling of the continuous speech signal; due to normal speech variability, two sequences arising from two utterances of the same word may exhibit a number of local differences.

These local differences may be that one element has been substituted for another, that an element has been inserted, or that an element has been deleted. Other local difference models are conceivable, for example, one that allows expansion of a single element into several elements or compression of several elements into a single element, as independent types of local difference. Given two sequences and costs (or weights) of the local differences, an alignment is assigned a cost equal to the sum of the costs of the local differences in it; the distance between the two sequences is the least cost of any alignment.

DTW is essentially a two-stage process; figure 1.9 illustrates the first stage. Two abstract speech-like patterns are shown, one vertically and one horizontally. Each pattern has time frames consisting of 3-element vectors; the vertical pattern has four frames, and the horizontal has five. The matrix in the centre is known as the "distance matrix" and it contains numbers which corresponds to the distance between each frame in one pattern and each frame in the other pattern. For example, the number "20" in the top right hand corner indicates that the first frame of the vertical pattern is quite different to the last frame of the horizontal pattern. Similarly, the "1" in row-2 column-2 indicates that the second frames of each pattern are very similar. The distance is actually calculated by taking the sum of the squares of the differences between each pair of frames.

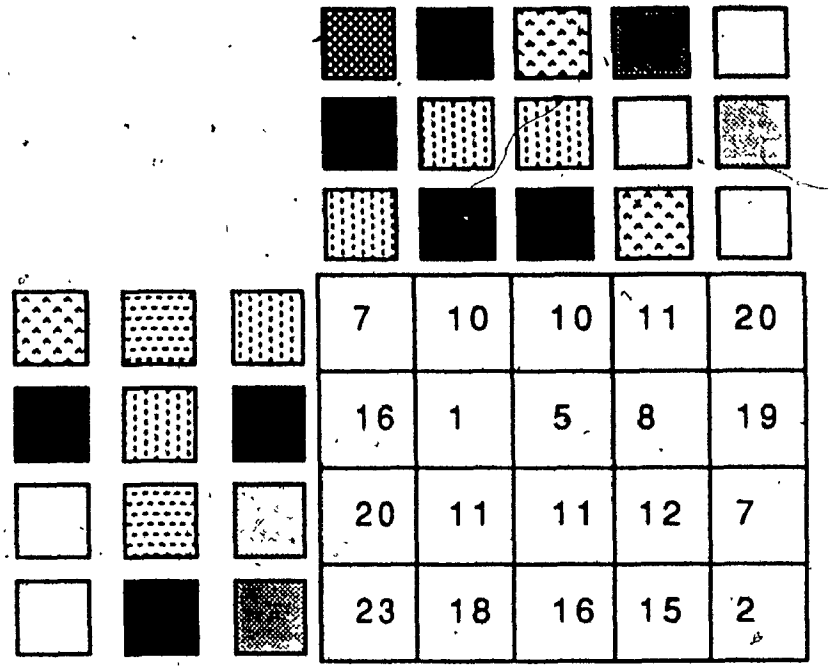


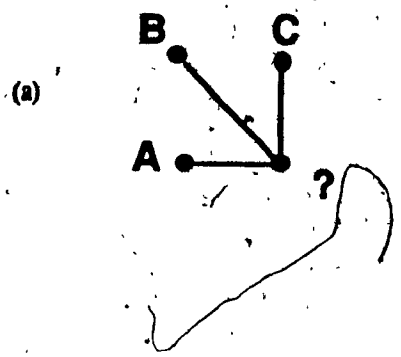
Fig. 1.9 Distance matrix obtained after comparing two abstract patterns [37].

The second stage is to find the path through the distance matrix, from the top left hand corner to the bottom right hand corner, which has the minimum accumulated sum of distances along its length. This path is the required non-linear relationship between the timescales of these two patterns, and it is found by dynamic programming.

Dynamic programming involves the regular application of a local optimization procedure which ultimately leads to an overall global solution. In this case a "local decision function" is used, together with the distance matrix, to construct a second matrix called the "cumulative distance matrix". Figure 1.10 illustrates the process. The local decision function is shown in figure 1.10a, and it defines that a path may arrive at any particular point either vertically, horizontally or diagonally. It is applied as follows:

For each point in the cumulative distance matrix, add the cheapest cost of getting to that point to the cost of being at that point, and enter it in the matrix. The cheapest cost of getting to a point is the smallest of the values in the previous entries (as defined by the local decision function) and the cost of being at a point is simply the value taken

from the corresponding position in the distance matrix. Hence, if this process is applied iteratively, starting at the top left hand corner of the matrix, it is possible to complete all the entries in the cumulative distance matrix.



(b)

7	17	27	38	58
23	8	13	21	40
43	19	19	?	

(c)

7	17	27	38	58
23	8	13	21	40
43	19	19	25	28
66	37	35	34	27

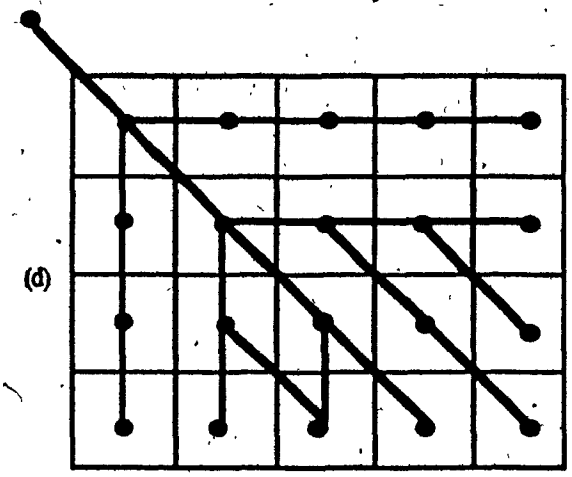


Fig. 1.10 Demonstration of Dynamic Time Warping; (After R.K. Moore [37])
(a) Local decision function, (b) Partially completed cumulative distance matrix,
(c) Completed cumulative distance matrix, and (d) Record of local decisions

Figure 1.10b shows the cumulative distance matrix in the process of being filled in. The "?" indicates the point being considered, and the three previous points are highlighted. The cost of getting to the point is the minimum of 19, 13 or 21, and the cost of being at that point is 12 (from the distance matrix in figure 1.9). Hence the cumulative distance entered at that point is 25 (13+12).

Figure 10c shows the cumulative distance matrix completely filled in. The number in the bottom right hand corner is highlighted because this is the overall distance between the two patterns; it is the sum of distances along the least-cost path through the distance matrix. To find the path it is necessary to remember at each point in the calculation exactly which local decisions were made (horizontal, vertical or diagonal). Figure 10d shows all of these decisions and it can be seen that they form a tree radiating from the top left hand corner (where the calculation started). The actual minimum cost path is found by tracing back along the local decisions, starting at the bottom right hand corner (where the calculation ended).

Referring back to the distance matrix (figure 9), the calculation shows that the least-cost path takes the route 7+1+5+12+2; no other path has a cumulative sum less than 27.

The formulation for this dynamic programming is the following recursive expression:

$$D(i,j) = d(i,j) + min\,[D(i-1,j), D(i-1,j-1), D(i,j-1)]$$

where $1 \le i \le I$ and $1 \le j \le J$ (I and J are the numbers of frames in the two patterns being compared), d is a distance measure between two frames, and the initial condition is $D(0,0) = 0$. The overall distance between the two patterns is $D(I,J)$.

Dynamic programming techniques originally developed for isolated word recognition have also been applied to the problem of recognizing connected words. Here, the spoken input is a sequence of words from a specified vocabulary, and matching occurs against isolated word reference templates. We are given an input pattern with some number of time frames. We also possess a set of reference templates, where each template has a length equal to the number of frames in that template. The goal is to find that sequence of templates that best matches the input pattern for a particular match criterion. The concatenation of templates is referred to as a "super" reference pattern. Two proposed solutions to this problem can be found in the two-level algorithm of Sakoe [41] and the level-building algorithm of Myers and Rabiner [42]. Also worth

mentioning in this context is the one-stage dynamic programming algorithm of Ney [43].

A brief description of Myers and Rabiner's level-building dynamic time-warping (DTW) algorithm for connected word recognition is as follows. The underlying idea is that the matching of all possible word sequences can be performed by successive concatenation of reference patterns. At the beginning, the time registration of the test pattern against a given super reference pattern is considered; it is observed that the algorithm can be implemented in levels, that is, one reference (of the super reference pattern) at a time. The computation matches test frames only against frames within a particular reference; the set of accumulated distances between different segments of the test pattern and that reference are saved and used as a set of initial distances for the next level. This idea is then extended to a level-building algorithm with multiple reference patterns, that is, when each reference of the super reference pattern is one of a set of reference patterns.

The recognition performance of isolated word recognizers based on DTW techniques is significantly better than that obtainable from linear time-normalization. This is because DTW provides a far more realistic timescale compensation process; greater variability can be accommodated, hence larger vocabularies may be used. Also by using relaxed endpoint constraints (the position where the timescale registration path is allowed to start and end), DTW does not suffer from the same dependency on endpoint detection as linear time-normalization. Hence the segmentor can be much simpler, and it is left to the DTW process to decide precisely where the words begin and end.

Votan is a commercial isolated word recognition system (vocabulary size of 256 maximum) which uses DTW for alignment.

1.3.2 Network based Systems

In the previous section we studied dynamic time-warping systems originally developed for isolated word recognition and later extended to recognition of strings of connected words. In this section we look at two representative network-based systems, CMU's Harpy system and IBM's Markov modeling system, which are directed toward the more difficult problem of continuous speech recognition. In the general form of this problem we are interested in large-vocabulary, speaker-independent recognition; the two systems under consideration restrict the problem considerably by introducing

grammatical and/or task constraints so that a simple finite-state model may be built of the entire language to be recognized.

Both systems compile knowledge at different levels of the language model into an integrated network. In the Harpy system, phonetic, phonological, lexical, and syntactic constraints have been combined into a single model which generates all acceptable pronunciations of all recognizable sentences; in the IBM system, each word of the top-level language model is replaced by a phonetic subsource, and then each phone is replaced by an acoustic subsource, yielding a model of all acoustical realizations of sentences in the language. An important difference between the two networks is the fact that, in the IBM system, all sources and subsources are Markov models, while in Harpy, Markov networks have given way to transition networks with no a priori probabilities associated to symbols that label transitions; as already mentioned, in both cases the integrated language models are finite-state models.

Another important difference is that Harpy uses segmentation while the IBM system does not. In Harpy, the acoustic signal is divided into variable-length segments that represent "stable" portions of the acoustic signal; spectral characteristics of each segment are then determined for use in phone template matching. The assumption here is that, given enough allophone templates, it is reasonable to attempt labeling of segments using pattern matching techniques. Asynchronous segmentation is performed top-down and then the network is used to select prototypes to be matched with the data. In the IBM system, no attempt is made to try to segment the speech into phoneme-like units: instead, a time-synchronous acoustic processor produces parameter vectors computed from successive fixed-length intervals of the speech waveform. A parameter vector coming from a 10-msec frame is matched against a set of prototypes: the parameter vector is then labeled by giving it the name of the prototype to which it is closest. Another possibility is that of using the input vector for retrieving a priori probabilities of different labels.

The Harpy system is an attempt to combine the best features of the Hearsay I system and the Dragon system [44]. The most significant aspects of the system design are an *integrated network* language model (knowledge representation) and use of *beam search* through the network during recognition. Segmentation is attempted, phonetic classification depends on unique templates, and word juncture knowledge is an integral part of the network. A word network exists such that any path through the network gives an acceptable sentence. Each word is replaced by a pronunciation network which represents expected pronunciations of the word. After words have been replaced by their

sub-networks, word juncture rules are applied to the network to model phone string variations due to influences of neighboring words.

During compilation into the composite network, various optimization heuristics are applied to yield an efficient phone network, that is, a network of acceptable pronunciations. During the recognition process, Harpy attempts to find an optimal sequence of phones satisfying two criteria: (a) the sequence must represent a legal path through the network and (b) the sequence should consist of phones with high acoustic match probabilities. It is possible that the best fit to a particular segment in the left-to-right search does not correspond to the correct interpretation: to compensate for this, a beam-search strategy is used in which a group of near-miss alternatives around the best path is examined. When the end of the sentence is reached, the phone sequence with the lowest total distance is selected; backtracing through the globally best sequence obtained at the end of forward searching yields the desired phone and word assignments.

A pronunciation dictionary and phone characteristics allow us to replace words with their sub-networks. A simplified sub-network for the word "was" is shown in Fig. 1.11. As before, redundant paths are removed; phonetic symbols are taken from the ARPAbet [45].

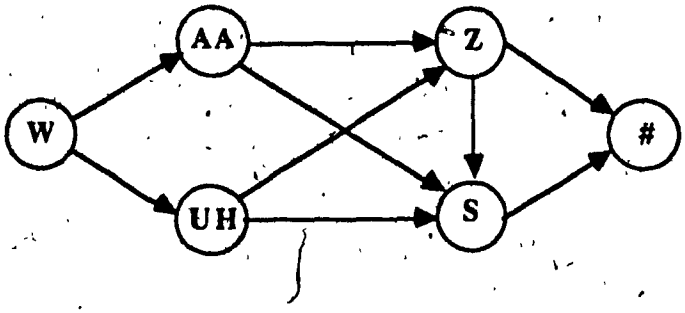


Fig. 1.11 Nonredundant Phonetic Network for word "was" [38]

So far we have seen illustrations of syntactic knowledge and lexical knowledge, although information about phone duration has been deliberately omitted from the latter. The phonetic network attempts to capture intraword phonological phenomena; word

boundary phonological phenomena, on the other hand, are represented by word juncture rules, which contain examples of insertion, deletion, and substitution of phones at word boundaries. The word juncture rules, which are then applied to the network as it exists so far. Finally as before, redundant paths are removed.

Harpy's front end performs rough segmentation based on zero-crossing rates and peaks in smoothed and differenced waveform parameters called the "zapdash" parameters. Quasi-stationary segments derived from the zapdash parameters are matched against phone templates. These phone templates are linear prediction spectral templates, and comparison is based on Itakura's minimum prediction residual error measure, which computes similarity in spectral shape. The spectral templates are talker specific but new templates may be learned automatically, for example, by adapting speaker-independent templates. The beam-search strategy for searching the finite-state graph prunes from further consideration paths scoring less than a variable threshold: rather than using a priori probabilities to find the most likely path through the network.

In systems like Harpy and Hearsay II [46], segments are detected asynchronously and then labeled: labeling a variable-length segment consists of recording, for each allophonic template, the probability that the segment represents an occurrence of that particular template. In contrast, synchronous nonsegmenting systems consider successive fixed-length frames of the speech signal. For each frame, we obtain a vector x of parameters representing the spectrum of that frame of speech. In vector quantization, our problem, for each such x, is to find that codeword x_i in a codebook of stored prototypes whose spectral distance from x is minimum. In this speech coding technique, we have the collection of possible reproduction vectors $x_1, x_2, \ldots, x_n$, which is stored in the reproduction codebook or simply the codebook of the quantizer; the reproduction vectors are called codewords (or templates). Moreover, we have a distance classifier which allows us to compare vectors according to a spectral distance. The encoding is illustrated in Fig. 1.12. Problems in constructing a good vector quantizer include choosing a good set of spectral representatives in the codebook, usually through training. More details about vector quantization can be found in [47].

The IBM Systems

IBM has developed two benchmark systems. The first one is a speaker-trained continuous speech recognition system with a recognition of 91% on words contained in

sentences in the 1000 word vocabulary set [48], [49]. The second system was an isolated word recognition system with an accuracy of 95% on a 8000 word office correspondence vocabulary [50]. This system has recently been enhanced by expanding the vocabulary to 20,000 words [51].

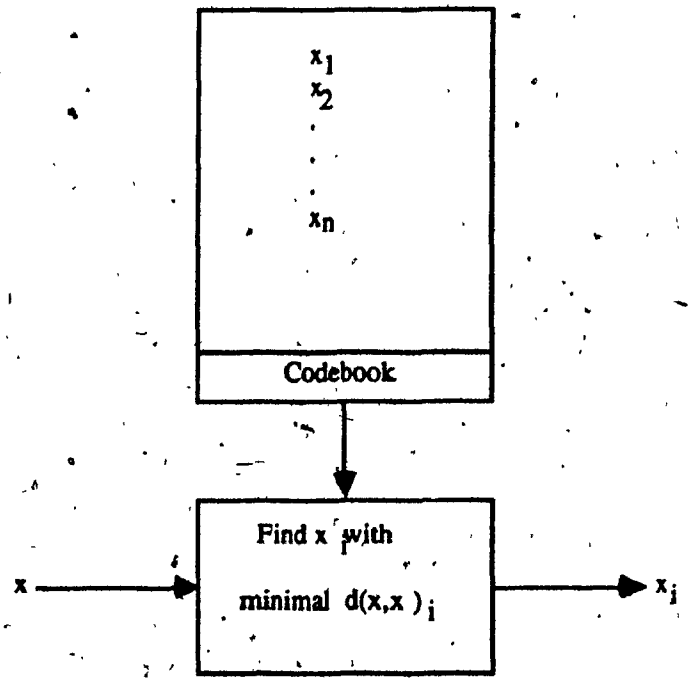


Fig. 1.12 Vector Quantizer Encoder [38]

The system is based on Markov models of language and has been implemented using two control strategies: (a) a Viterbi algorithm and (b) a left-to-right stack decoder algorithm that estimates the probability that a given partial hypothesis can be extended to yield the actual sentence. Important aspects of the system design include the presence of a priori transition probabilities in the finite-state language model and formulation of speech recognition as a problem of maximum-likelihood decoding. As such, statistical models of the speech production process are required. The choice between the two control strategies or decoding methods mentioned earlier is a function of degree of task constraint, the size of the state space.

In the IBM approach, the allowed sentences are either described *a priori* by an artificial grammar or else limited by a vocabulary and a task domain in which models may be constructed from observed data. The distinctive feature of the IBM approach is

that speech recognition is formulated as a problem in *communication theory*. The speaker and the acoustic processor are conceptually combined into a single unit, the *acoustic channel*. Fig 1.13 shows the relation between the text generator, the acoustic channel, and the linguistic decoder.

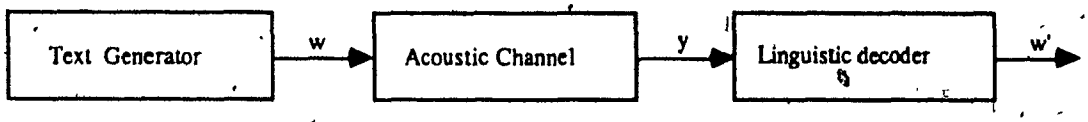


Fig. 1.13 Speech Recognition as a Communication problem [38]

In Fig. 1.13, w is a string of words generated by the text generator, y is a string of acoustic processor output symbols (more specifically, a string of prototype identifiers, one for each 10 msec of speech), and w' is the word string produced by the linguistic decoder as an estimate of the word string w. The acoustic channel provides the linguistic decoder with a noisy string from which it must attempt to recover the original message. The linguistic decoder searches for a word string w which maximizes the probability $P(w,y)$ of the joint observation of (w,y) at the two ends of the channel. A stochastic model of the acoustic channel will account for both the speaker's phonological and acoustic-phonetic variations and for the unvarying performance of the acoustic processor. Given models that specify both $P(w)$ and $P(y|w)$, the linguistic decoder may determine w using some algorithm which is appropriate to the size of the language.

The model of text generation is a language model. Both the language model and the acoustic channels are Markov sources consisting of states connected by transitions; with each transition there is an associated output word. A probability is attached to each transition. More details on Markov models will be explained in Chapter 4.

In the IBM system, the language model assigns probabilities to strings of words. For the acoustic channel model for single words, a phonetic Markov subsource is associated for each word. The possible output strings, drawn from an alphabet of phones, are all the different phonetic pronounciations of the word. An example is shown

in Fig. 1.14. For each word there is a set of *phonetic subsources* and for each phone there is a set of *acoustic subsources*. An acoustic subsource for a phone is a Markov source whose output alphabet contains the output symbols of the acoustic processor and which specifies both possible acoustic processor outputs for each phone and their probabilities. More details on stochastic decoding and its performances can be found in [52], [53].

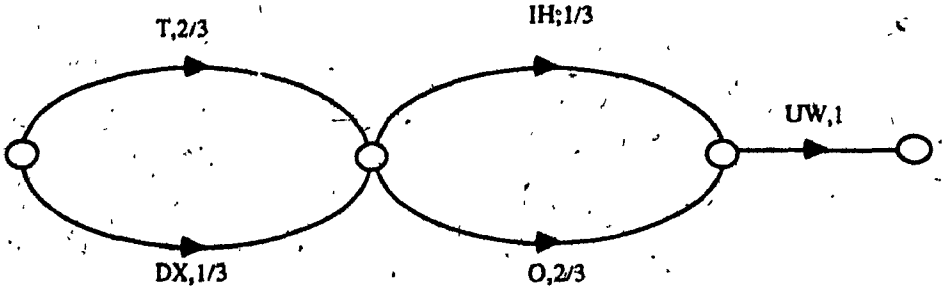


Fig. 1.14 Phonetic subsource for the word "two" [38] (Phones in ARPAbet notation)

Results obtained on stochastic model based network system shows that there is no significant difference in recognition accuracy from that of DTW approach. However, from a computational point of view the Markov models require an order of magnitude less storage and execution time where the DTW based techniques have a very simple training phase (only data collection) and a very complicated recognition phase, Markov models are just the reverse. It has been overwhelmingly agreed that Markov models provides the correct balance for any practical system [37].

1.3.3 Knowledge-based Systems

The purpose of a "good" recognition model is to take knowledge and generalize that knowledge appropriately to assess previously unseen events. This is possible only

by a proper understanding of the variabilities involved.

The conclusion drawn after the ARPA project on SUR was that there still exists a great need for integrating more speech knowledge into ASR models in order to solve difficult tasks as well as for achieving better recognition results.

The performance of an automatic speech recognizer ultimately depends on the amount and quality of the training material. However, if the dimensionality of the representation is raised, then recognizers are always going to be undertrained. It is therefore vital to know how the knowledge embedded in the training material can best be structured and hence utilized. In theory, it ought to be possible to extract a great deal of structural information from the speech signal itself since humans can do it. So the question is how to obtain more knowledge from speech data.

The main characteristic of knowledge is that it is highly domain-dependent. Abstractly speaking, knowledge is made up of descriptions, relationships, and procedures corresponding to a given domain of activity. In practice knowledge can take many diverse forms. It roughly consists of "objects" and their relationships in a domain together with the procedures and heuristics for manipulating these descriptions.

It is obvious that the error-prone nature of speech data makes it necessary to have an efficient cooperation between highly diversified knowledge sources: knowledge concerning phonetics, phonology, prosody, lexicon, syntax, semantics, and pragmatics.

The choice of adequate structures for representing the available knowledge sources is a crucial problem in speech understanding as well as in any AI systems. Several approaches were taken in the past which includes several interesting ideas.

One possible solution is to use a single structure in which all the diverse knowledge sources are integrated. This was the solution chosen in the Harpy system. Harpy integrates knowledge of all levels in a precompiled network which contains the various phonetic transcriptions of all syntactically legal sentences. The only disadvantage of this approach is that the size of the network becomes too large and storing all possible sentences causes the system to be too rigid.

A second solution is the other extreme as that of Harpy, a total independence of the various knowledge sources. A heterarchical method for implementing such a scheme called the blackboard model was used in the Hearsay II system [54]. Fig. 1.15a shows an example of a blackboard organization. In this approach the knowledge sources are independent processes which, in principle, are not aware of each other and which asynchronously post hypothesis at various levels (phoneme, syllable, word, etc.) to a global data base called the blackboard. This way a sentence is described at different

levels. Invocating a given knowledge source is data-directed in the sense that specific preconditions must be fulfilled to access the blackboard. Such blackboard schemes were successfully applied to various other AI areas like vision [55] and signal interpretation.

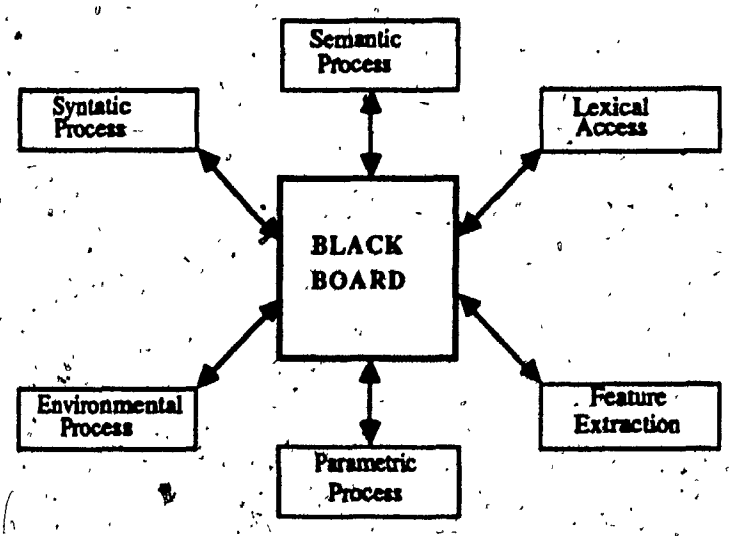


Fig. 1.15a Example of a Blackboard Model

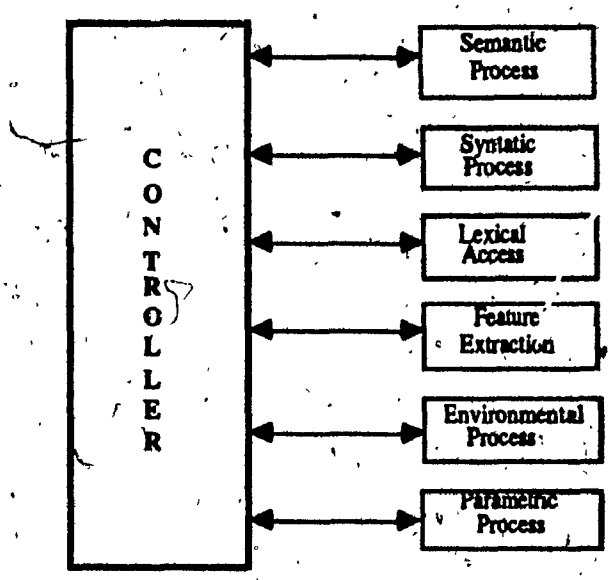


Fig. 1.15b Example of Hierarchical Structure

A third solution, which is an intermediate approach between the network model and blackboard model, is the hierarchical model as shown in Fig. 1.15b. In this approach the processing is controlled by some kind of control structure or supervisor. In contrast with the data-driven, asynchronous activation of Hearsay II, the knowledge sources in this model are activated by the supervisor. This way control strategies can be tested by modifying the supervisor. The Hwim system of BBN was developed based upon such an approach.

Rule-based Expert Systems

Recent results obtained in AI are largely due to sophisticated problem solving systems called expert systems. For a well defined and restricted domain, expert systems are able to reach the level of expertise of a human expert.

Instead of the classical two-tiered organization of data and program, expert systems introduce a more flexible, three-level structure, data, knowledge base, and control [56]. Fig 1.16 shows an illustration of the overall organization of an expert system.

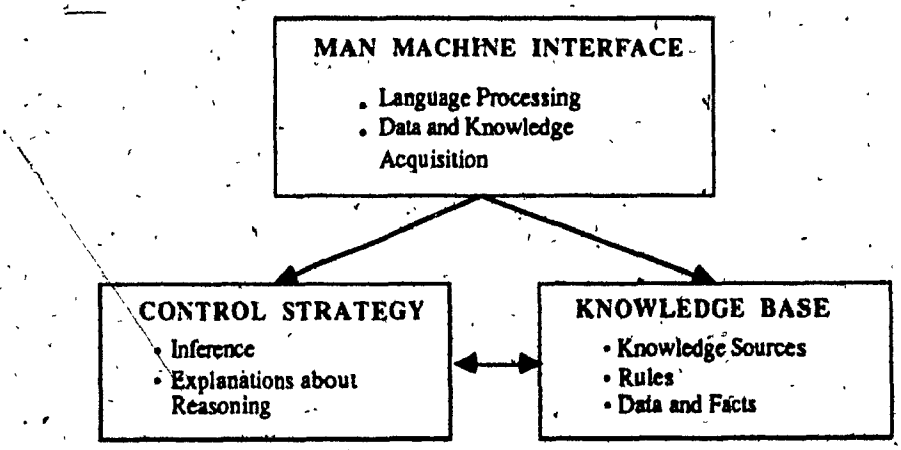


Fig. 1.16 Organization of an Expert System [39]

The knowledge base is used by the system for analyzing the problem in a deductive way. It typically incorporates some kind of data-activated operators whenever specific preconditions are met during the problem solving process.

Expert systems make it possible to split a complex expertise into a large number of relatively simple rules. A human expert often seems to use a production rule scheme while reasoning. Therefore, these systems can be succesfully applied to various aspects of speech recognition that require solving specific and limited problems.

Some of the attempts made so far which uses expert systems approach are in speech spectrogram interpretation, multi-expert structure for accessing large lexicon. In the multi-expert system, different experts in the society execute in parallel various algorithms derived from a task-decomposition of the speech recognition algorithm.

The major difference between Hearsay IIs blackboard model and the expert society lies at the level of the control structure. In Hearsay II the only way the various knowledge sources (the experts) can communicate is by asynchronously posting hypothesis in a data base. The knowledge sources are triggered when specific preconditions in the data base are satisfied. In the expert society each expert is provided with a specific control strategy and communicates directly with other experts. This strategy makes use of planning algorithms and is related to the AI concept of frames which provides an interesting framework for the predictive use of knowledge. An example of this can be found in [68].

1.4 Problem Statement and Thesis Objectives

We have seen several approaches to speech recognition each using varying amount of speech knowledge and different ways of knowledge representation. Template matching techniques use constraints to define a manageable task and are easy to develop, however, they use very little speech knowledge. The Harpy system used more speech knowledge integrated into its network model. The IBM system showed techniques for combining speech knowledge with well defined mathematical models. Such techniques did manage to take into consideration speaker variations and coarticulation effects to a certain extent. The Hearsay II and Hwim systems showed the importance of using independent knowledge sources even though both used different types of knowledge representation. Expert systems were shown to be promising for knowledge representation in a natural way and they are especially attractive if the domain is specific, small, and well defined.

Most of the past work on ASR clearly demonstrates that the solution of difficult speech recognition tasks involving speaker-independency, connected speech or large vocabularies need much more speech knowledge, knowledge from all levels, in their models. Faster and larger computing systems may help in solving this problem. It is also understood that speech signals contain sufficient knowledge since humans seem to process it very easily.

In order to integrate knowledge from different sources which are of different nature and which are available at different levels, one should adopt different types of knowledge integration techniques, by applying the most appropriate ones at each level, rather than clinging to one specific model, in their recognition models.

The primary task of this research work is to explore the possibility of integrating more speech knowledge into a speech recognition system for solving a difficult task: speaker-independency. The emphasis is more on finding properties which are invariant to inter- and intra-speaker variations.

The speech recognition system under development is based upon the Procedural Network (PN) for knowledge integration. The PN system is a hierarchical model and it is capable of integrating knowledge from other levels in the hierarchy. Therefore the PN system has a touch of various techniques which were incorporated into several other successful systems. Details of the Procedural Network system are given in Chapter 2.

The procedural network can be considered as a supervisor of the complete system. The model consists of several key components in which each component solves a specific task. A task can be of any type, for example, deciding whether a given sound is "nasal" or "plosive". Each such task is considered as a sub-network in the PN system.

The main contribution of this thesis work is to develop a sub-network to solve a particular task. The task involves identification of vowels and diphthongs for any given speaker and for any given context. The sub-network which solves this task, called the Speech Pattern Analyzer (SPA), uses some vision analysis techniques on speech spectrograms. The details of this approach is elaborated in Chapter 3 and Chapter 4 of this thesis. In Chapter 3 we show how speech spectrograms can be treated as patterns and how perceptual organization can be applied on these patterns for obtaining morphological descriptions. In Chapter 4 we illustrate how stochastic models like Markov chains can be used to learn certain morphological properties in which the parameters are in frequency domain rather than traditional time domain.

Chapter 5 explains the incorporation of several pieces of knowledge pertinent to the characterization of a vocalic region. Word candidates are scored based upon measures

which are dynamically adjusted depending on detected properties. This approach attempts not to rule out any likely candidate while discarding least likely candidates. Each candidate which has a score is returned to a higher level in the PN hierarchy.

An example of the application and performance of the Procedural Network is illustrated in Chapter 6. The test data base consists of isolated letters and digits collected from a large population of male and female speakers whose mother tongue is not necessarily English. Final recognition results as well as some intermediate results from the SPA sub-network are also included in this chapter.

Chapter 7, the last chapter, gives some discussions about the underlying philosophies and the goals achieved. Some suggestions regarding extending the vision approach as a signal interpreter into other areas of speech as well as into other areas of signal interpretation is also mentioned in this chapter.

Chapter 2

A Procedural Network Based Speech Recognition System

A paradigm for automatic speech recognition using networks of actions performing variable depth analysis is presented. The paradigm produces descriptions of speech properties that are related to speech units through Markov models representing system performance.

2.1 Introduction

Recent results on Automatic Speech Recognition (ASR) and Speech Analysis suggest that progress in designing recognition devices and in advancing speech science knowledge may arise from an integration of the so called cognitive and information-theoretic approaches [44].

The cognitive approach attempts to infer analytic knowledge about possible speech invariants and their relations. Work by Zue [57], Klatt [58], Stevens [59] and De Mori et al. [60], [61] are along this line.

The information theoretic approach is based on a performance model containing states and transitions between any pair of states [52]. Probabilities can be learned that the system is in any of the model states or is changing state through any of the allowed transitions. Furthermore, the model generates in each state or in each transition, observable system parameters or descriptors according to some statistical distribution.

Our model attempts to integrate both of the above mentioned approaches into one model and applies it for solving a difficult task in ASR, speaker independency.

The idea is that of extracting speech properties using knowledge about acoustic correlates of linguistic units. For an abstract linguistic unit, the corresponding acoustic

correlates have attributes which may differ from an instantiation to another due to the fact that different speakers produce different signals even if they intend to pronounce the same sound. Attribute statistics of sound properties collected on a large variety of pronounciations of the same sound from different speakers are probably the best knowledge we can gather today for characterizing different speaking styles. Furthermore, some expected acoustic properties can be missed in some cases and some unexpected properties can be detected in some other cases. These aspects can also be characterized by stochastic performance models.

An important problem arising when large vocabularies have to be recognized is that of identifying a possibly small set of Speech Units (SU) with which all the possible words and word concatenations can be obtained by compilation. The learning problem is then reconducted to the conception of a performance model for each SU.

The model proposed in this work has a data-driven component that identifies Acoustic Segments (AS) based only on acoustic evidence and knowledge of the information bearing properties corresponding to different spectral structures. An Acoustic Segment usually contains at least a vocalic part identified by resonances represented by narrow band spectral lines in a time-frequency-energy representation of speech. An AS may contain one or more vowels with one or more consonants in the "vocalic" part. An AS may also have a head and a tail. Heads and tails may contain low energy sonorant consonants or consonants characterized by frication noise or by a transition between a deep dip in the signal energy curve and an energy peak.

The coarse acoustic properties that characterize ASs and are used for delimiting them are by no means interpretations , they are just elements for focusing the attention of the property extractors. Relations between ASs and SUs like phonemes, diphones or syllables are established by performance models representing, for each SU of interest, insertions, deletions, substitutions and their statistics.

As unambiguous segmentation of continuous speech into ASs is very useful for reducing the complexity of word hypotheses generation and verification because word hypotheses can start only at specific time instants of the head, vocalic part or tail of an AS.

For the head, vocalic part and tail of each AS, plans of property extraction operators (procedures) are executed. These plans produce descriptions of speech segments. These descriptions may apply to segments of variable duration. Ferguson [8] has shown how performance models can be built under the assumption that properties generated by a model have variable duration. Plans producing descriptions perform a

sort of Variable Depth Analysis because they may generate different types of properties for different segments.

Unfortunately there is no suitable theory for the conception of performance models in such a case, but an interesting research is in progress [62]. Such research is motivated by the existence of a similar problem in different application areas [63], [64]. While waiting for a probabilistic theory of hetereogeneous pattern descriptions, a pseudo solution can be adopted where local decisions are made by a sort of "intelligent vector quantizer" that generates, for each AS a string of symbols belonging to a vocabulary VH of hypothesis competitions. Such a vocabulary contains phoneme symbols, and more general phonetic classes indicating partially ordered sets of competing phonemic hypotheses. Hidden Markov Models (HMM) for SUs are built which generate symbols of VH. Such HMM may include segmental duration statistics and can be compiled to represent any natural language word vocabularies or sentences.

Another approach consists in simply multiplying probabilities or scores obtained in each segment even if acoustic properties may vary from segment to segment.

The validity of this approach has been tested on a multispeaker task consisting the recognition of sequences of letter and digits with a short pause between any pair of them.

2.2 A model for Computer Perception of Speech

The speech signal $x_1(t)$ is generated by a discrete and finite sequence of actions,

$$A = a_1(t_1)a_2(t_2)a_3(t_3)\ldots a_k(t_k)\ldots a_K(t_K), \qquad (1)$$

where $a_k(t_k)$ denotes an action ending at time t_k ; $a_1(t_1)$ represents the silence preceding the beginning of a sentence.

When a person reads a sentence S, a relation

$$R_1(S, A) \qquad (2)$$

is applied which produces A. The relation R_1 may depend on the speaker, his/her mood, state of health and history. As R_1 may produce several AS for the same S, probability distributions for all the possible AS can be derived using a generative model.

The speech signal $x_1(t)$ is generated by the sequence of actions A using another relation,

$$R_2(A, x_1(t)) \tag{3}$$

R_2 depends on the anatomy of the speaker. Again, the same actions may produce different signals, because the speech production system is soft and its behavior is affected to some extent by the environment.

If the speaker does not read but generates a sentence from a set C of concepts, then a third relation is applied:

$$R_3(C, S) \tag{4}$$

R_3 may depend on the speaker and his/her culture. Statistical models can also be used for characterizing this relation.

The generation of $x_1(t)$ can be seen as the application of the following composite relation:

$$G = R_3 \circ R_1 \circ R_2 \tag{5}$$

according to the scheme shown in Figure 2.1.

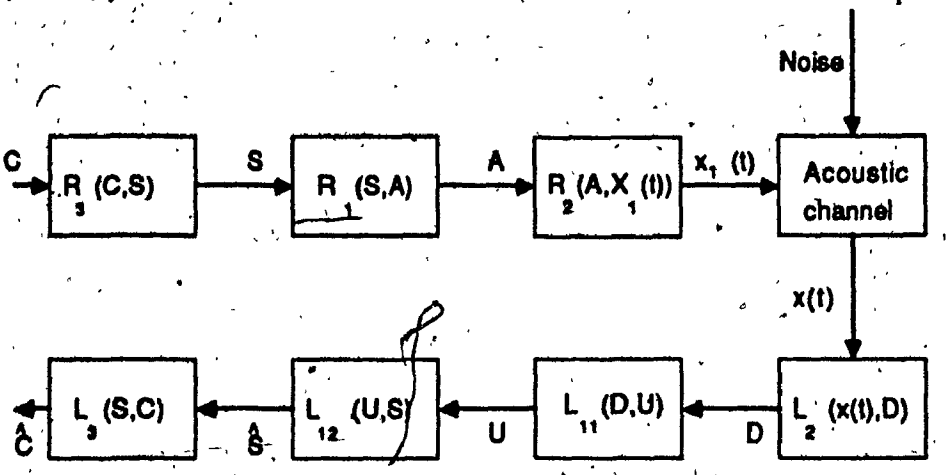


Fig. 2.1 Speech Communication Channel

Recognition consists in applying the relations in the opposite direction. Unfortunately we have only a limited knowledge of these relations. We have used it for building speech synthesizers. We do not even know the alphabet $\Sigma_A \{a_k\}$ although we

know alphabets Σ_C and Σ_S for the elements of C and S respectively. Furthermore, signal $x_1(t)$ is affected by noise and is transformed into $x(t)$ through the acoustic channel.

As we do not know Σ_A, nor we know R_2, we can characterize actions by descriptions of what they produce. According to this approach, the perception of $x(t)$ consists in extracting a sequence of descriptions:

$$D = d_1(\tau_1)\, d_2(\tau_2) \ldots d_i(\tau_i) \ldots d_I(\tau_I) \tag{6}$$

where $d_1(\tau_1)$ describes the silence proceeding the beginning of $x(t)$ and $d_i(\tau_i)$ describes the segment of $x(t)$ between the time instants τ_{i-1} and τ_i.

Segments of D can be 10 msecs. frames ,or intervals of variable duration obtained by a segmentation algorithm like the one proposed in [20].

The descriptions D can be obtained by perceptual actions in analogy with the generative scheme. Perceptual actions, as well as generative actions, have to be defined and used according to a criterion of economy. That is, there must be a limited number of actions (operators) based on which a variety of networks of actions can be built.

Recognition can be seen as a combination of a relation

$$L_1(D, S) \tag{7}$$

that is the perceptual counterpart of relation R_1 used for speech generation, and a relation:

$$L_2(x(t), D) \tag{8}$$

The relation $L_2(x(t), D)$ is deterministic in the sense that it can produce only one description D for a signal $x(t)$. Description D can be of fixed duration , i.e. a descriptive phrase is generated at constant time intervals, or of variable duration , i.e. descriptions can be generated for intervals of different length. If we want to maintain the analogy with the production model just outlined, D should be of variable duration because the articulatory actions (gestures) are of variable duration.

Descriptions must refer to parameters, morphologies and properties that are characteristic for a sound and exhibit low variances when many speakers, different microphones and environments are considered.

In practice, fixed duration models have been developed and tested with a considerable degree of success mostly in speaker-dependent systems. In one of the most

successful systems developed so far [65]. D is a sequence of symbols obtained every 10 msecs. by vector-quantization with a process that is speaker-dependent and context-independent.

Relation $L_1(D, S)$ has to capture two different types of knowledge. The first type of knowledge is a relation:

$$L_{11}(D, U) \tag{9}$$

between a sequence U of Speech Units (SU) and corresponding descriptions D. There are speech units like the plosive sound /b/ for which a large variety of different descriptions D are perceived as the same sound. Relation L_{11} is many-to-one and it could be interesting to collect statistics of the elements of the universe of acoustic descriptions that produce the perception of the same linguistic sound. These statistics may represent distributions of acoustic patterns produced by a single or many speakers having the intention of producing the same sound. Statistics may also take into account characteristics of background noise.

The choice of SUs, for our purpose, has to be based on practical considerations as well as on theoretical ones. For example, for some purposes, units can be just phone classes or syllable classes.

The introduction of SUs is important because once a vocabulary Σ_u of speech units has been chosen and effective relations between each $SU \in \Sigma_u$ and descriptions of acoustic properties have been established, large varieties of word and sentence models can be built by compiling networks of SU models.

A second type of knowledge is a relation:

$$L_{12}(U, S) \tag{10}$$

where S is a linguistic entity such as a sentence and U is a sequence of Speech Units. L_{12} can also contain statistics. L_{12} may represent how different speakers may have different pronounciations of the same word. A stochastic model representing a word W in terms of SUs can be built.

An interesting possibility, which is attempted in this work, is that of designing L_2 and L_1 procedurally, through actions to be performed on $x(t)$ in order to obtain D, U and S.

Knowledge-based extraction and interpretation of signal properties has proven to be very effective when interpretation can benefit from contextual relations [66].

Descriptions D of different level of detail (depth) can be obtained depending on

model expectations or the already available context.

It seems that variable depth descriptions can be very useful in complex tasks where a preliminary selection of hypotheses has to be done based on robust but simple descriptions and then a more detailed analysis has to be performed involving levels of depth depending on the competing hypotheses.

The entire perception model can be represented by procedural networks which invoke subnetworks at several levels. At each level, different types of units can be defined and statistics of their components can be collected.

Building procedural networks is an activity of conditional planning. Waldinger [67] discusses elementary planning techniques and conditional planning. An introduction to the use of planning techniques for ASR is described in [68].

The most general procedural network has to operate along two dimensions using acoustic properties extracted in different time intervals and at different levels of detail. It will be shown in the following sections how these capabilities can be implemented.

In order to perform variable depth analysis, a context in which the analysis is performed has to be defined. Algorithms were proposed in the past for segmenting continuous speech into Pseudo-Syllabic-Segments [20]. Although these algorithms have shown good performances in different tasks and for varieties of speakers, they were not error free in segmenting the speech signal into syllables. The principal reason for these errors was that segmentation was based only on acoustic evidence. Acoustic Segments (AS), obtained by segmentation algorithms based on acoustic properties, have to be treated as data rather than interpretations. Being based on acoustic evidences, ASs can be used for driving and delimiting the extraction of more detailed acoustic properties.

Following the above considerations, relation (9) can be represented by a Procedural Network (PN) expressing U as a sequence of words and each word in terms of ASs. Other PNs are then introduced for expressing how the components of D, extracted from each AS, relate to the Speech Units of U.

Fig. 2.2 shows a first level PN representing a relation between (U = zero) and its Acoustic Segments. Fig. 2.2 is a sort of performance model for a segmentation algorithm described in [20] and based on Primary Acoustic Cues (PAC) whose definition is recalled in Table 2.1. In this particular example, each AS invokes procedural knowledge for each of the vowels and consonants contained in it. Segmentation of the word "zero" may produce two ASs with probability P_{001} or one AS with probability P_{002}.

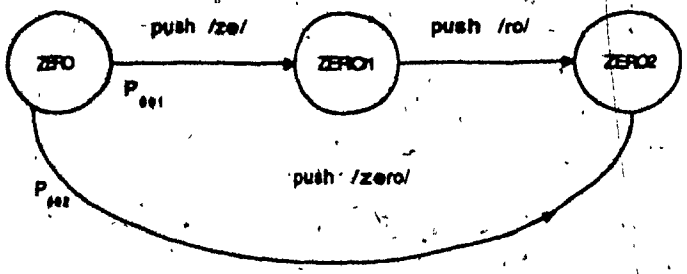


Fig. 2.2 Performance model of the segmentation of the word /zero/ into AS.

Each arc label in Fig. 2.2 corresponds to the invocation of a PN that will evaluate the scores of the Speech Units contained in the AS. These scores could be a-priori probabilities computed by Markov sources. The presence of two different paths in Fig. 2.2 would suggest using separate sub-sources for "ze" and "ro" and a single sub-source for the entire word if acoustic properties show evidence of a tight coarticulation between /e/ /r/ and /o/.

Figure 2.3 shows the loudness curve and the time evolution of other acoustic properties for an utterance of the word "zero".

A PAC description of the curves is also shown in Fig. 2.3. Acoustic Segments are delimited by an asterix.

2.3 Procedural Networks

A Procedural Network (PN) can be described with a formalism similar to that used for an Augmented Transition Network Grammar (*ATNG*). This formalism has been successfully used for Natural Language and Pattern Recognition [69]. A PN is a 5-tuple,

$$PN = [i, Q, A, q_0, q_f] \tag{11}$$

where, i is the network identifier, Q is a finite set of states, A is a finite set of directed

Table 2.1 Primary Acoustic Cues

Symbol	Attributes	Description
LPK	tb,te,ml,zx,	long peak of total energy (TE)
SPK	"	short peak of TE
MPK	"	peak of TE of medium duration
LOWP	"	low energy peak of TE
LNS	tb,te,zx	long nonsonorant tract
MNS	"	medium nonsonorant tract
LVI	tb,te,ml,zx	long vocalic tract adjacent to a LNS or a MNS in a TE peak
MVI	tb,te,ml,zx	medium vocalic tract adjacent to a LNS or a MNS in a TE peak
LDD	emin,tb,te,zx	long deep dip of total energy
SDD	"	short deep dip of total energy
LMD	"	long dip of total energy with medium depth
SMD	"	short dip of total energy with medium depth
LHD	"	long non-deep dip of total energy
SHD	"	short non-deep dip of total energy

Attribute Description

Attribute	Description
tb	time of beginning
te	time of end
ml	maximum signal energy in the peak
emin	minimum total energy in a dip
zx	maximum zero-crossing density of the signal derivative in the tract

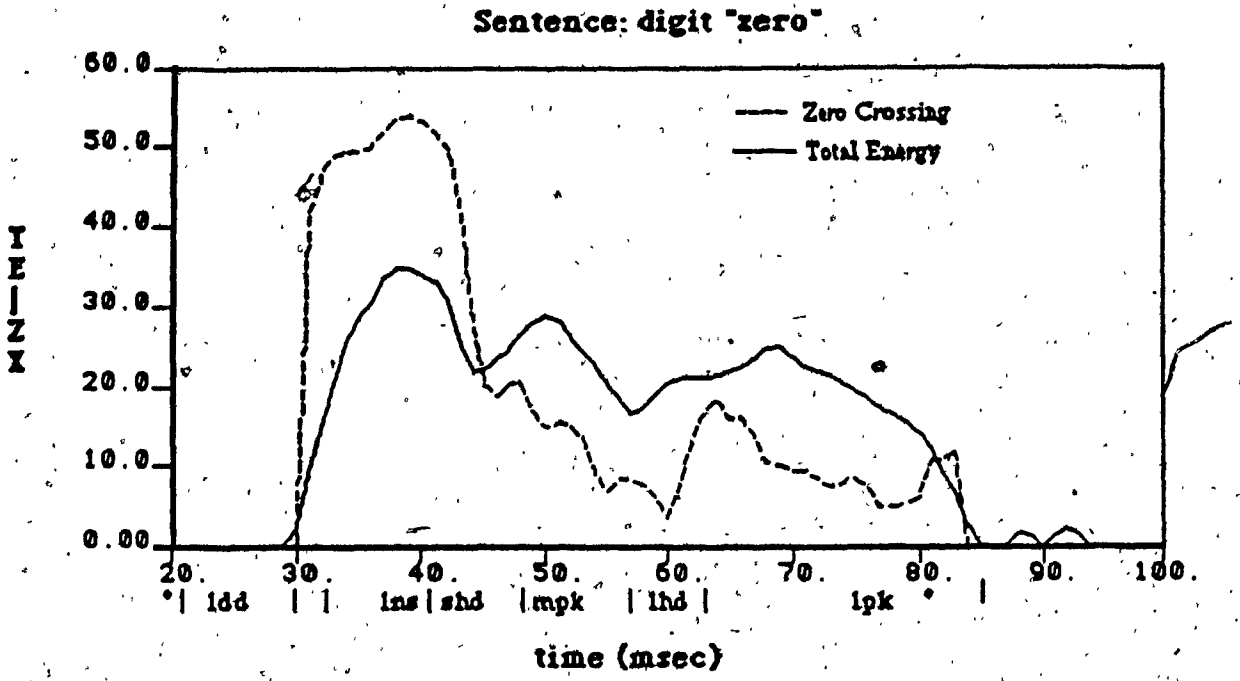


Fig. 2.3 Loudness and other acoustic parameters used for obtaining the PAC description of an utterance /zero/.

arcs; $q_0 \in Q$ is the initial state and q_f is the final state. Without any loss of generality we consider only PNs with a single initial state and a single final state.

Each arc, $a_i \in A$ is a 5 - tuple:

$$a_i = (q_b, q_e, P_i, condition_i, action_i) \qquad (12)$$

where $q_b \in Q$ is the starting state of a_i, $q_e \in Q$ is the terminal state of a_i, P_i is a measure associated to the arc (it can be a weight or a probability according to the scoring method used by the PN supervisor described later on), $condition_i$ is a condition and $action_i$ is an action; both of them are associated to the arc. The conditions can be categorized in two classes:

COND n

refers to a user defined condition *n*.

DEFAULT r

refers to a default condition (it is satisfied only if no other condition of any arc whose starting state is q_b returns a measure greater than *r*).

The actions are executed by the PN supervisor and can be categorized in five classes:

EXE n

executes a user defined action; such an action is usually a "matcher" which performs some computations on the input data and returns a result.

PUSH i

is defined in the following manner. Let's assume that PN_j has an arc that contains PUSH i. Let π_j be the process that executes PN_j. When the arc is reached whose associated action is PUSH i, the execution of π_j is suspended. The state of π_j is pushed on the top of the stack of the PN supervisor. A new process π_i that executes PN_i is created and executed. When the final state of PN_i is reached, the last arc of PN_i is considered. It has associated either a POPABS f or a POPCOND f action. This action is executed. It returns scores computed by PN_i. These scores are passed to π_j whose execution is resumed while π_i terminates.

POPABS f

is associated to the final state of a PN. It stops the execution of the current network process and the result of the execution of the user defined function f is returned.

POPCOND f

This action is also associated to the final state of a PN. It stops the execution of

the current network if all the paths in the network leading to the final state have propagated their contribution to the computation of the scores the PN has to provide. If the condition is reached, then the result of the execution of the user defined function f is returned.

JMP

makes the score associated to q_b propagate to q_e without any change.

A procedural network is a formalism for implementing knowledge representations by compiling concatenations of a limited number of basic actions. These actions produce scored interpretations of segments of the speech signal that have to be combined with scored expectations represented in data structures attached to network actions. Plans made of action sequences are scheduled by the supervisor agenda that uses composite scores for arranging priorities. Scheduling may take into account other parameters related to the resources and their availability.

Each PN is associated with a Working Memory (WM). Actions associated with the arcs of a subnetwork produce descriptions stored into the subnetwork WM. When a push to a subnetwork is made, the supervisor may link the subnetwork WM with other WM, thus establishing the viewpoint within which conditions are tested. Most of the actions associated with arcs include plans, Hidden-Markov-Models (HMM), local parsers, and rule-based inference units. All these tools are used for extracting an unambiguous description D of a speech pattern and for computing an a-priori probability under a hypothesis H:

$$P(D/H) \tag{13}$$

The PN supervisor keeps up to date a search space where each node is represented by the following four-tuple:

$$(q, context, T, score) \tag{14}$$

where:

- q is a state of the PN, with a buffer containing the information propagated by the actions executed before reaching it,
- *context* is the context (viewpoint) in which the conditions and actions of the arcs starting at q have to be executed,
- T is the starting time of the speech signal for the execution of sensory

procedures invoked by the actions associated with the arcs starting at q,

- *score* is the score of the hypothesis contained in or implied by *context* up to T; score could be $P(D(t_0 T)H)$ where t_0 is the beginning of the sentence. T could also be a set of possible time references; in this case, score will be a set of scores $\{s(t) | t \in T\}$.

- Composite scores can be evaluated as likelihoods:

$$L(D,H) = Pr(D/H)Pr(H), \tag{15}$$

where $Pr(H)$ is obtained by a language model.

The size of the search space can be kept small in spite of a large number of states in the PN if conditions and actions are properly chosen and placed in the network.

Example 2.1

An example of a PN is shown in Fig. 2.4. This PN is invoked by a PUSH /ze/ in Fig. 2. 2.

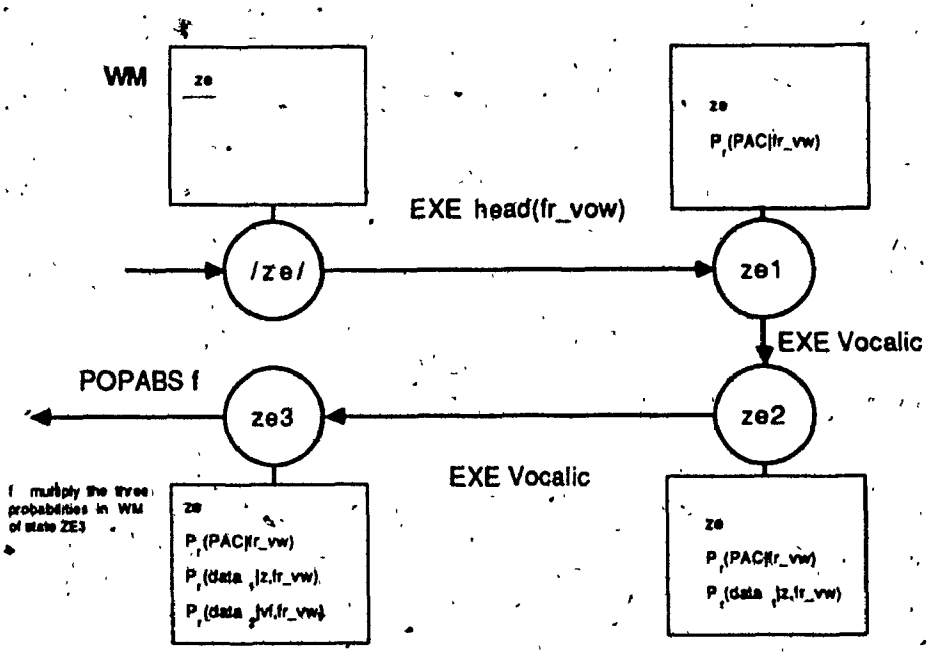


Fig. 2.4 An example of a Procedural Network (PN)

The PN is supposed to extract data from the AS under analysis and to produce a score that is an estimation of the a-priori probability that the data extracted from AS have been observed during the pronounciation of /ze/.

The first state is associated a WM containing the hypothesis /ze/. The first arc is associated an EXE action whose function consists using an HMM for the syllabic class "fricative-vowel" (fr-vw). This HMM generates PAC symbols. The statistics of the process it represents have been obtained by the Forward-Backward (FB) algorithm on strings of PACs corresponding to ASs obtained by the pronunciation of /ze/ from many speakers. Action EXE HMM (Pac (fr-vw)) returns the following score:

Pr (PAC (AS under analysis) / fr-vw).

This score is stored into the WM when state ze1 is reached.

The arc starting at ze1 has associated the action EXE head (fr-vw). The function head (fr-vw) extracts the head of the AS under analysis and executes a "Network of Actions" (see [68] for details) on the segment head for computing the acoustic properties that are relevant for discriminating among fricative sounds. The syllable class *fr-vw* acts as context in this case. Let $data_1$ be the properties extracted from the AS head. Properties $data_1$ are related to the SU /z/ and the following score is computed:

$$Pr\ (\ data_1 / z, fr\text{-}vw\ )$$

The above probability could be obtained directly or through the probabilities of the place and manner of articulation for /z/.

In the case of the following vowel, the SU /e/, there is no need to hypothesize the manner of articulation in order to distinguish /ze/ from the other syllables of the language to be recognized. Thus, only the properties $data_2$ for the place of articulation are considered and the score (vf stays for front-vowel):

$$Pr\ (\ data_2 / vf, fr\text{-}vw\ )$$

are computed by the action EXE vocalic. The details of action EXE vocalic is explained in the following chapters of this thesis.

Eventually, the final state ze3 is reached and the action POPABS f is executed. The associated function f for computing the cumulative score is:

$$f: compute\ Pr\ (data/ze) = Pr\ (Pac/fr\text{-}vw)\ .Pr\ (data_1 / z, fr\text{-}vw).\ Pr\ (\ data_2 / vf, fr\text{-}vw\ )$$

There are some problems made evident by this example.

The first problem is that PNs like the one of /ze/ and the one of /se/ have a lot of actions in common whose execution should not be duplicated. A more efficient network organization is presented in the next Section.

The second problem is that different networks may extract different types of data in the same AS making score comparisons rather difficult.

This problem can be avoided by organizing the PN hierarchy in such a way that properties for an AS are extracted only by a single action.

The third problem is that properties like $data_1$ and $data_2$ extracted in two different intervals of the same AS may be different in number and in quality. This problem is common to other Pattern Recognition applications [62], [63], [64]. A pseudo-solution of it consists in just multiplying probabilities of different segments. Other solutions are under investigation.

A fourth problem is that the boundary between heads, vocalic parts and tails may be fuzzy. This possibility is not considered in the application described in this system although it will be worth investigating.

2.4 The Elaboration - Decision paradigm

An Elaboration-Decision (ED) paradigm is executed through a certain number of ED cycles, which analyze descriptions of the same acoustic segment at different levels of depth. ED cycles are procedures. According to this approach $d_i(\tau_i)$ in (6) is made of components extracted at different levels of depth. Thus, $d_i(\tau_i)$ can be represented as follows:

$$d_i(\tau_i) = d_{i0}(\tau_i)\, d_{i1}(\tau_i) \ldots d_{ij}(\tau_i) \ldots d_{iJ}(\tau_i) \qquad (26)$$

J being a function of i : $J = J(i)$.

Description $d_{i0}(\tau_i)$ is made of Primary Acoustic Cues (PAC). These cues are extracted by a spontaneous activity.

The first ED cycle executes an Elaboration Phase (EP) that computes a first level description $d_{i1}(\tau_i)$. A Decision Phase (DP) is executed that uses as data the following description:

$$d_i(\tau_i) = d_{i0}(\tau_i)\, d_{i1}(\tau_i) \qquad (27)$$

DP scores all candidate hypotheses written in the active working memory of the ED-action. If hypothesis scores are such that there is only one candidate with a high score, then the ED process keeps only that candidate and terminates. Otherwise, other ED cycles are performed until the scores of the best-scored candidates are different enough or all the possible ED cycles have been executed.

Local scores are a-priori probabilities. For example, the score of hypothesis H in τ_i is:

$$Pr(d_i(\tau_i)/H) = Pr(d_{i0}(\tau_i)/H)\, Pr(d_{i1}(\tau_i)/d_{i0}(\tau_i).H)) \qquad (28)$$

Different ED-actions may extract different data and obtain different descriptions for the same time interval τ_i.

There are possible solutions to this problem. The first one, which has been adopted in this model, consists in using homogeneous descriptions (PAC sequences) for selecting the best path of actions. Each one of these paths contains only one ED-action for each signal subsegment. Thus the probabilities computed for each subsegment are based on the same data for all the considered hypotheses.

Another pseudo solution, that could be investigated, consists of letting all the ED-actions score hypotheses about every possible candidate and then computing, for each candidate, the probability of all the properties extracted by different ED-actions in the same signal interval.

This is equivalent to assuming:

$$d_i(\tau_i) = \prod_{j=1}^{J} d_{ij}(\tau_i) \qquad (29)$$

where $d_{ij}(\tau_i)$ is the description extracted by ED-action (j) in τ_i.

Another problem arises from the fact that local probabilities may not be "homogeneous" in different intervals τ_i because they have been computed on descriptions extracted at different levels of depth. There are different possibilities for combining local probabilities. One of them consists in "summarizing" the states of each

local conflict of hypotheses by a symbol belonging to a summary Alphabet Σ_{SA} and using strings of summaries for building a performance model. Such a performance model can be a Markov Source. A model for each word can be constructed and used together with a Language Model for computing sentence likelihoods as in [52]. The other possibility, that has been adapted for the "letters-and-digits" protocol, consists in just multiplying the probabilities of a sequence of segments in order to obtain a cumulative score. This problem too needs further investigation.

A detailed description of the properties used in the ED paradigm for recognizing letters and digits will appear in other papers describing motivations for choosing certain properties and the experimental conditions in which statistics have been collected [68], [70]. A brief summary of these properties is given in the following.

2.4.1 Plosive sounds

Various properties extracted both in the time and frequency domain have been used. They are described in [68].

The first description level consists of PAC. The second description level contains buzz-bar and burst indicators extracted from the time waveform, its envelope, the time evolution on certain frequency bands.

The third description level contains burst properties extracted in the time intervals in which burst indicators were detected.

The fourth description level is related to spectral line transitions at the voice onset.

2.4.2 Vocalic intervals

Spectral lines are extracted with an algorithm described in [34]. Markov Models are used for modeling statistics of frequency and energy of spectral lines. They generate probabilities of place of articulation for intervals of 20msec. duration. Significant changes in these probabilities are used for detecting possible subsegments. Sequences of hypotheses are considered in a tree search procedure in which a node corresponds a sequence of labels like:

$$(v_1 : fw, v_2 : fw, v_3 : bw, v_4 : bw)$$

where f stays for "front", b for "back" and w for "vowel". The node sequence corresponds

to the hypothesis that subsegments v_1 to v_4 represent a sequence of a front vowel followed by a back vowel. The product of the probabilities of each subsegment hypothesis is used as score for the node. These properties and other vocalic properties detected are described in detail in Chapters 3, 4, and 5 of this thesis.

2.4.3 Other consonants

Liquid and nasal sounds are hypothesized using a mel-scaled filter bank and considering time evolutions of energy differences in a continuous parameter Markov Model. Other levels of descriptions involve the use of Markov Models for spectral lines in the stationary zone of the consonant and in the transient segments in order to capture statistics of properties discussed in [60].

Fricative sounds are hypothesized using a continuous parameter Markov Model whose parameters are the energies of another special mel-scaled filter bank suitably designed for analyzing time evolutions of mid and high frequency energies. More details on the content of this subsection can be found in [71].

2.5 The Supervisor

Several strategies can be applied in order to build a state space of hypotheses and to find the most plausible one. A Dynamic Programming approach has been used to design the PN supervisor for the application outlined in Section 2.4.

Let a_i be an arc of the j^{th} PN. Let s_k be the k^{th} segment of the input speech signal. The contribution of such an arc is,

$$c_i = f_i(p_i, g_i(cond_i), h_i(act_i)) \tag{30}$$

where:

p_i is the score associated to a_i, g_i is a function which returns the evidence of satisfaction of condition i, h_i is the function which returns the value computed by the action i and f_i is the function which combines the values of its arguments in order to give the contribution of the arc.

According to the definition of p_i, g_i, h_i, and f_i, several interpretations of the contribution of an arc are possible.

The assumption made for the experiment described in this work is summarized in the following. Score p_i is the a priori probability of an arc; g_i is the probability that the condition is satisfied; h_i is the probability that the segment s_k matches with the knowledge used by action act_i; f_i is a multiplication operator. The contribution c_i can be rewritten as:

$$c_i = p_i\, g_i(cond_i)\, h_i(act_i) \tag{31}$$

Let $s = s_1, s_2, \ldots, s_n$ be an input sequence of speech segments, $PN_k = \{K, Q, A, q_0, q_f\}$ be the k^{th} PN and $a = a_1, a_2, \ldots, a_n$ be a sequence of arcs in the network PN_k such that the initial point of a_1 is q_0 and the terminal point of a_n is q_f.

We would like to find the sequence of arcs "a" which maximizes the conditional probability,

$$P(a/s) = P(s/a)/P(s) \tag{32}$$

for a given s. That is to find the sequence "a" which maximizes:

$$P(s/a) = \prod_{i=1}^{n} \left[P_i g_i(cond_i) h_i(act_i) \right] \tag{33}$$

Given a fixed value of k, $k \in \{1,2,\ldots, n-1\}$

$$\max_{a} P(s/a) = \max_{q \in Q} \left[\max_{q_k} P(s/a^k) \right] \tag{34}$$

where $a_k = a_1, a_2, \ldots, a_k, a_{k+1}, \ldots, a_n$ is a sequence of arcs such that the terminal point of a_k and the initial point of a_{k+1} is q (*i.e.* a_k is the sequence of arcs such that the corresponding sequence of states contains q as the k^{th} state of the sequence). The DP algorithm computes the second term of (34).

2.6. Chapter Summary

- A theory of machine preception of speech based on actions and a paradigm for its implementation based on procedural networks is proposed.
- Variable-depth analysis, the possibility of invoking subnetworks at

different levels and of using Markov models by some subnetworks are among the novelties of the proposed approach with respect to previously proposed network-based models for ASR [72], [73].

- The possibility of using acoustic properties as in [74] with stochastic models of their descriptions is another novelty of the proposed approach.
- It has been shown how the hypothesize-and-test paradigm already used or proposed for ASR and lexical access [75], [76], [77], [78] can be implemented with procedural networks.
- Performances of the tests are given in full detail in Chapter 6.

Chapter 3

Speech Spectrogram Interpretation: A Problem In Biological Vision

3.1 Generalities

Speech Spectrograms, a 3-dimensional time-frequency-energy representation of speech signals, has been in existence for the past four decades or so. Since then, speech spectrograms were the most valuable aid used by researchers for the Recognition and Understanding of human speech. Latest experiments by Zue and Cole [79] further highlights the significance of using spectrograms in ASR research. According to Zue and Cole, in contrast with previous beliefs, acoustic signals contain a great deal of phonetic information, and this information can be extracted from sophisticatedly represented speech spectrograms.

According to early work done on spectrograms [79], [80], [81], [82] it has been established that the underlying phonetic information can be recovered entirely by visually examining the spectrogram. An experienced spectrogram reader can correctly identify close to 90% of phonetic segments by visual examination. However, as it stands now, no machine can perform such tasks with such a degree of accuracy. Using spectrograms, one could work in a visual domain rather than in the conventional audio domain. Working in visual domain may be better because it is easier to "verbalize" speech spectrogram process than to verbalize hearing process. Speech spectrogram readers interpret spectrograms based on several a priori knowledge, namely

a) by considering pictorial scene changes
b) by relating acoustic knowledge and phonetic knowledge to these scene changes
c) by associating prior linguistic knowledge to such scene changes

There have been several knowledge-based systems developed or under development in order to read and interpret speech spectrograms [80], [83], [84], [85].

The SPEX system of Verbex [83] is a speech spectrogram Expert. SPEX has three expert modules, namely, the vision expert, acoustic-phonetic expert, and the phonetic expert. The vision expert extracts symbolic features from the spectrogram which are then interpreted by the acoustic-phonetic expert. The acoustic-phonetic expert generates phonetic hypothesis based on already detected symbolic features. Finally, the phonetic expert reasons about allowable phoneme sequences which leads to the final recognition.

SPEX's idea was to give more emphasis on qualitative and symbolic reasoning rather than quantitative number crunching.

Another system was developed by Zue and Lamel [80]. In their approach, the expert system tries to extract acoustic features from spectrogram. The acoustic feature belongs to predefined feature set. The knowledge for extracting these features was composed of a set of rules which were highly "qualitative" in nature. For example, features were described as present/absent, and associated with values like high/medium/low or weak/strong. The acoustic features includes, voice on-set, location, strength etc. The system was tested only on Stop consonants and reported results close to that of an expert human spectrogram reader.

A third system called SONEX was developed by Stern, Eskenanzi, and Memmi [84]. Their approach was to detect phonemes directly from a presegmented spectrogram. The system tries to interpret phonemes from segments using acoustic knowledge provided as rules. The rules themselves were highly "quantitative" in nature.

For example, consider the rule to recognize the fricative sounds, /f/ and /s/:

IF NOISE (IN-SEG-3)
AND NOISE_LOW_LIMIT(IN-SEG-3) ϵ [2100,2500] Hz
THEN PHONEME(IN-SEG-3) IS (+40) /f/, (+10) /s/

SONEX system was tested for a preselected set of phone candidates from 7 different phonetic class as follows:

/t/, /k/	⇒ voiceless stops
/d/	⇒ voiced stop
/f/, /s/	⇒ voiceless fricatives
/z/	⇒ voiced fricatives
/n/	⇒ nasal stop
/R/	⇒ liquids and glides
/i,a,u/	⇒ vowels

Both Zue's system and SONEX have somewhat similar approach. Both systems do not take into account the temporal changes which appear in spectrograms which is a very significant feature and the feature extraction process for both are highly quantitative. Furthermore such systems do not consider knowledge about distributions of properties across a speaker population.

3.2 Speech Pattern and Vision Techniques

In our approach, the 3-dimensional speech spectrograms are treated as images, and well known image processing (pattern recognition) techniques are applied on these images. Contextual variations in speech images are somewhat similar to real world changes in scene analysis. Therefore, by considering speech pattern as images and applying image processing techniques to these patterns, one should be able to interpret and capture variations in a better way. There are several reasons for taking such an approach.

Firstly, speech signal is highly variable. Existing ASR systems lack the power to understand unexpected variations, absence, or hidden features caused by inter-speaker variations. However, contextual variations can be observed on speech spectrograms, and therefore, by treating speech spectrograms as images, these variations can be better represented and recognized.

Secondly, existing ASR systems tend to look for specific features and take action based on the presence or absence of the expected features. These features could be at the acoustic level or at the phonetic level. It is very likely that a "misplaced" feature means absence of that particular feature. An example: there does not yet exist a formant

tracker that can determine formant frequencies reliably, especially in the regions where it is likely to find formant transitions as in the case of diphthongs [80]. In this approach, by keeping redundant information together with significant information, one does not throw away certain contextually significant features.

Finally, knowledge about spectrograms is incomplete. We know that some properties that can be visually detected are relevant for perception. The same property may have variations from pattern-to-pattern of the same word because of inter- and intra-speaker variations. It is important to characterize knowledge about such variations. This characterization has to be statistical because we do not have other types of knowledge on how basic word pattern prototypes are distorted when different speakers pronounce that same word. On the other hand, it is very important to characterize word prototypes in terms of properties that are relevant for speech production and/or perception.

Property based prototypes of words of speech units may differ from real patterns not only because properties are distorted, but also because some properties are missed or some unexpected properties have been inserted. Insertions and deletions can be often characterized by deterministic rules reflecting basic coarticulation knowledge, but in many cases they cannot be fully explained and are better characterized by statistical methods.

Based on the above considerations, the system proposed in this thesis represents an attempt to integrate knowledge-based extraction of relevant speech properties and statistical modelling of their distortion.

Before we look into the details of our approach, a brief review on some of the vision aspects are considered first. Some of the vision techniques which researchers established decades ago, motivated us to take such an approach.

3.3 Visual Recognition

Earlier research in computer vision assumed that, vision is made possible by matching the 3-dimensional knowledge of an object's appearence against some kind of corresponding 3-dimensional reconstructed image from the data. A recent work by Lowe [86] established that, visual recognition can commonly be achieved directly from the 2-dimensional image without any preliminary reconstruction of depth information. Lowe, in his work, proposes several non-traditional view points on computer vision which closely relates to biological vision. According to Lowe, the two major components

to be looked into for any vision recognition system are,

a) Spatial Correspondence,

b) Perceptual Organization, and

Spatial correspondence includes measured (observed) properties, such as, shape, color, texture, connectivity, context, orientation etc. More details on perceptual organization can be found in [87], [88], [89]. Spatial correspondence is established "when the measured locations of features in the image are in accurate agreement with the predicted locations of features for a particular projection of some known object" [86]. Emphasis is given to correspondence because, locational information is usually the strongest source of data in terms of number of measurements and accuracy of measurement, in the presence of "noise".

Perceptual organization is the basic capability of the human visual system to formulate relevent groupings from an image without prior knowledge. Human beings are capable of detecting Symmetry, Clustering, Collinearity, Parallelism, Connectivity etc. from any random image by simple observation. Most computer based vision systems do not incorporate most of the above mentioned capabilities, since, such groupings may not lead to immediate physical interpretations.

3.3.1 Importance of Perceptual Organization in Vision

The study of perceptual organization came from Gestalt Psychology during the 1920's and 30's. According to Gestaltists, "perception was something that happened as a whole rather than as a combination of individual primitive features" [90]. The word, Gestalt, itself means "whole". Based on the Gestalt theory on perceptual organization, Wertheimer [91], founder of the Gestalt school, demonstrated a grouping phenomena for perceptual organization. The grouping phenomenon were categorized as:

1. Proximity:	Elements that are closer together tend to be grouped together.
2. Similarity:	Elements that are similar in physical attributes, like, color, orientation, size, etc. are grouped together.

3. Continuity:	Elements that lie along a common line or curve are grouped together.
4. Closure:	Completed curves are grouped together to form closed regions.
5. Symmetry:	Elements that are bilaterally symmetric about some axis are grouped together.
6. Familiarity:	Elements that are seeing together are grouped together.

Fig 3.1 shows some examples on grouping discussed above.

Unfortunately, the Gestalt theory on perceptual organization faced a setback because of several reasons. The most important reason was the difficulty involved in deriving any type of quantitative theory to support the psychological viewpoints.

Recent work by Witkin and Tenenbaum [92] examines the role of grouping phenomena in both biological and computer vision. Human beings are capable of deriving structure and organization directly from a collection of 2-dimensional image features. Preliminary grouping can be made possible even without any high-level knowledge of the content of the image and these groupings are likely to stay intact throughout later stages of interpretation of image.

According to Witkin and Tenenbaum, the later interpretation merely consists of attaching labels to the already established primitive groups. Also, one of the strongest argument put forward by Witkin and Tenenbaum was that the strength of primitive level grouping process does not come from a-priori expectations but rather from a "non-accidentalness" argument.

Perceptual organization is so significant to the extent that it is unlikely to have happened accidently and, therefore, is likely to represent meaningful information of the scene. The degree of non-accidentalness, in turn, reflects the degree of significance. According to Lowe, some of the derivation methods for image organization are the following.

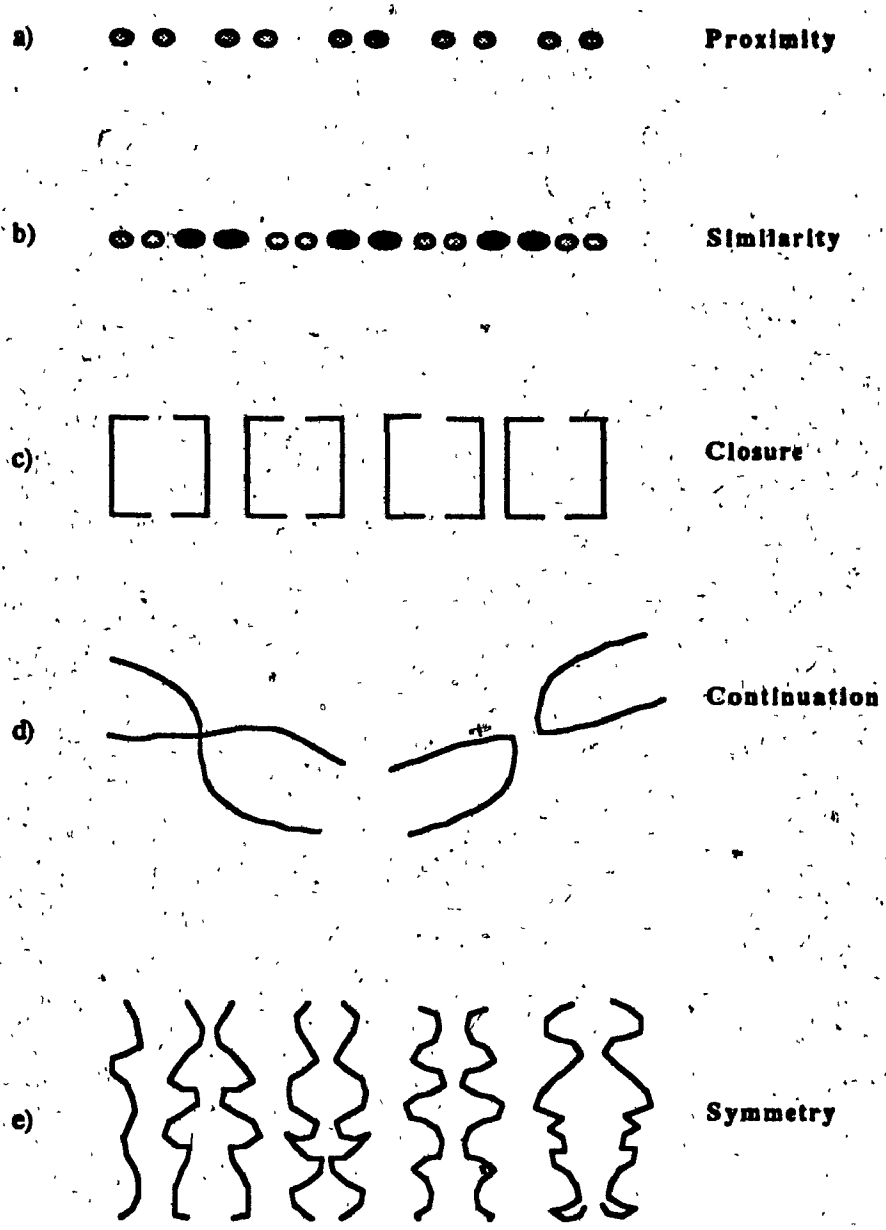


Fig. 3.1 Examples from some of the categories of Grouping Phenomena developed by Gestaltists [86]

a) dots are paired based on the basis of proximity;
b) dots are paired on similarity in size;
c) shapes are grouped as squares due to closure;
d) lines are seen as crossing due to good continuation;
e) bilaterally symmetric pairs of lines are grouped.

3.3.2 Calculating the probability of accidentalness

For doing so, one could associate the conditions such as, viewpoints invariance conditions, prior knowledge of probability of occurance, recursive application of structuring etc. Image relations are basically divided into two classes: those arising through an accident of viewpoint and those that are having meaningful structure in the scene. By calculating the probability of relations those are non-accidental in origin would tell us which relations are worth detecting and thereby evaluating their siginificance. The problem of calculating the probability can be subdivided as follows [86]:

1. Knowledge of the image projection process leads us to the conclusion that only certain class of image relations will occur more often than by chance and will therefore be statistically detectable.
2. Statistical estimates of non-accidentalness can also make use of prior knowledge of the probability of occurence of each relation.
3. The formation of the accidental instance can be modelled by assuming independence of position and orientations.
4. Initial relations can be recursively combined into relations that can influence the original estimates of significance.

3.3.3 Limiting Computational Complexity

Human vision fails to detect groupings in cases where objects are surrounded in a field of similar features. For example, many animal camouflages hide regularities in the animal's structure by surrounding them with nearby spots. The failure to detect highly significant structures seems to be clearly be a limitation of human vision rather than a functional feature [86]. The limitations of human vision are presumably the result of the inherent computational complexity which involves finding all possible significant relations in an image.

One method for limiting this complexity is to only examine groupings which consists of features that are close together in the image. A second possibility is to take all the features in a given region and to histogram them according to various properties and look for statistically significant peaks. This is the basis of texture analysis where

statistical methods are used for characterizing sets of features.

Image relations deals with only a few features at a time and are highly sensitive to feature's spatial location, but most texture measures treat an arbitary number of features within a given region without too much concern for the feature's locational details. The latter is not quite applicable in the case of speech images since locational information is very important.

What we have discussed so far is the general Gestalt theory on vision and its recent application in image recognition as pointed out by several researchers. It is very interesting to see how these theories may well fit into speech recognition problems by applying them to speech images like speech spectrograms.

3.4 Motivation to treat Speech Spectrogram Recognition as a problem in Vision.

Consider the speech spectrogram image shown in Fig. 3.2. If we look at the image and consider only 2-dimensions, time and frequency, one could observe quasi-parallel lines followed by curves and other such curves all over the spectrogram. If we include the 3rd dimension, the energy, also in our observation, one can see the strength and weakness of these lines and curves which are represented in terms of darkness and lightness. Some of the curves and lines are *accidental* in the image while some of them are *non-accidental*. Those appeared accidently are because of:

a) noise in speech signal
b) aspirations caused by the speaker
c) environmental noise
d) coarticulation effects

On the other hand, those lines and curves which are non-accidental in nature carry significant acoustic, phonetic, and linguistic level information. The initial task involved in recognizing such an image would be, somehow, to detect and interpret the information which is non-accidental in the presence of the accidental one.

It is obvious at this point that one could treat speech spectrograms as images and then it would clearly be a problem where biological vision techniques such as, spatial

correspondence and perceptual organization can be established. Expert speech spectrogram readers, knowingly or unknowingly, use such an approach while reading spectrograms. In Fig 3.1, one can easily observe perceptual group elements as well as spatial correspondence which is the locational information. Locational information is the strongest spatial information and is most important when quantitative measurements have to be associated in presence of noise. The basic catagories in grouping, such as, Proximity, Similarity, Concatenation, Closure, and Familiarity all can be used in grouping speech images for perceptual organization.

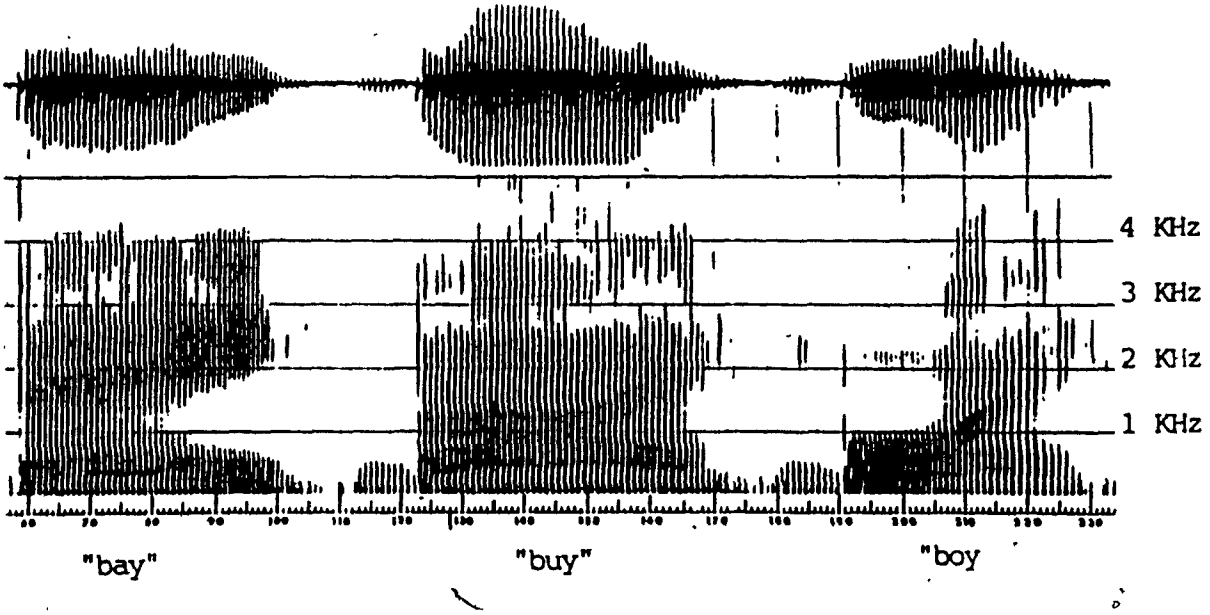


Fig. 3.2 A sample speech spectrogram of words "bay", "buy", and "boy"

3.4.1 Why treat Speech Spectrograms as Images ?

When it comes to the question of speaker independency, speaker variability is always the difficult problem in ASR. Past work in this area clearly demonstrates that a pure quantitative approach cannot deliver satisfactory results. Recently, researchers in this area have begun to consider using qualitative informations as well as quantitative [56], [81], [85] elements in speaker independent ASR problems. What are qualitative elements in speech? Qualitative components are exactly the same as spatially and perceptually grouped elements in vision. Spatial and Perceptual information can be associated to acoustic and phonetic level information in speech.

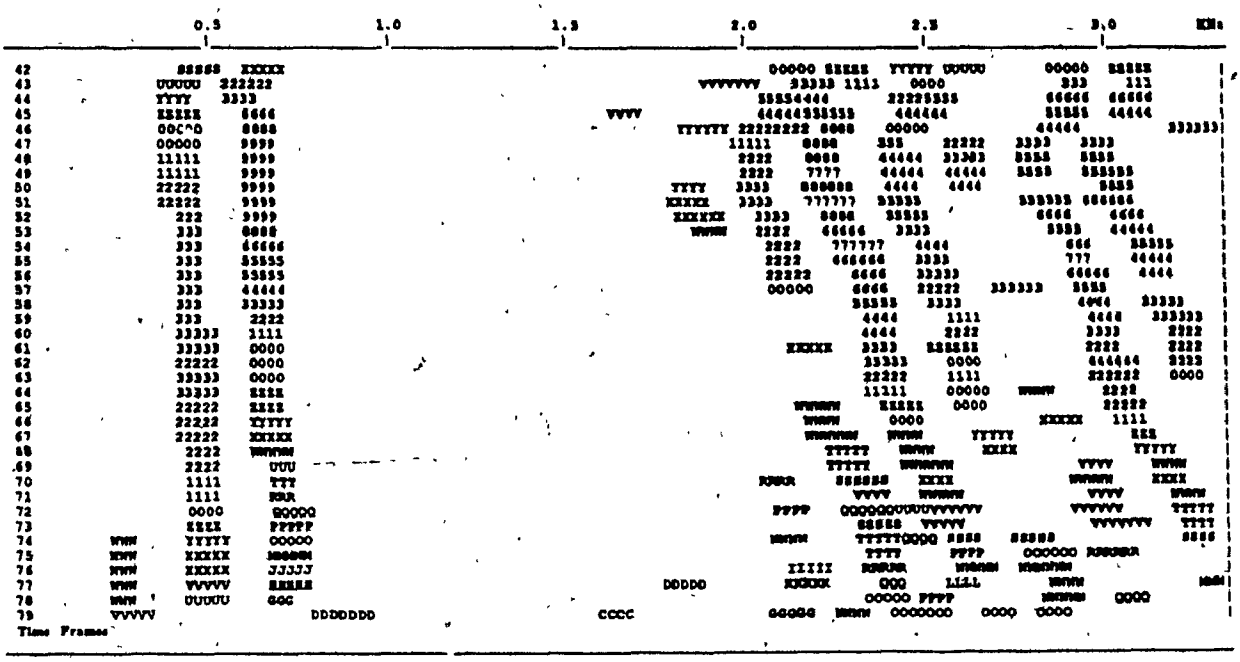


Fig. 3.3 Example of a speech pattern of letter "a" spoken isolated.

3.4.2 Generating Speech Images (Speech Patterns)

For conceptual reasons, from now on we will designate speech images as speech patterns. The time-frequency-energy pattern of speech segments are obtained by considering the 0-4KHz portions of spectra computed with the Fast Fourier Transformation (FFT) algorithm applied to the preemphasized speech signal. Fig. 3.3 shows an example of such a pattern.

In Fig 3.3, frequency from 0 to 3.5 kHz is shown along the horizontal axis and each printed line corresponds to a centisecond interval. Also in Fig. 3.3, letters indicate the peak relative energy. Letter "B" represents twice the energy that is represented by letter "A"; digit "0" represents an energy that is twice that represented by letter "Z" and so on. Digit "9" is the strongest energy point.

The pattern in Fig. 3.3 consists of elements which appear accidently and also which appear non-accidently. Accidental elements reflect noise etc., while non-accidental elements exhibit strong linguistic properties of the speech sound. Detection and isolation of accidental elements are not possible at this point. Therefore, all elements are considered to be non-accidental until proven to be accidental. Based on

the theory of Gestalts Psychology on vision, the speech pattern is considered as *a whole* and spatial and perceptual groupings are carried out on these patterns. Before applying grouping algorithms, the speech pattern is skeletonized and preprocessed for reasons which are explained below.

3.4.3 Skeletonization

It is important to notice at this point that, the smallest significant elements in speech patterns considered are the lines and curves and not isolated points. Lines and curves represent perceptual informations while locational information about lines and curves represents spatial correspondence. There are two important restrictions imposed on the choice of the skeletonization algorithm:

1. connectivity of lines should be maintained by keeping the points at junctions.
2. excess erosion shouldn't be allowed.

An existing algorithm called safe-point-thinning algorithm [93] did meet the above conditions and hence it was used.

There is an argument that skeletonization process would usually reduce information content in patterns and also that it would introduce spurious information. Skeletonization would also to some extent, introduce positional variations on patterns. However, since we are working on lines and curves as the smallest elements and not individual points, small variations or shift in lines and curves will not affect the grouping strategy.

Skeletonized patterns are easier for spatial description and for perceptual grouping. Fig. 3.4 shows the skeletonized pattern of the original pattern in Fig. 3.3.

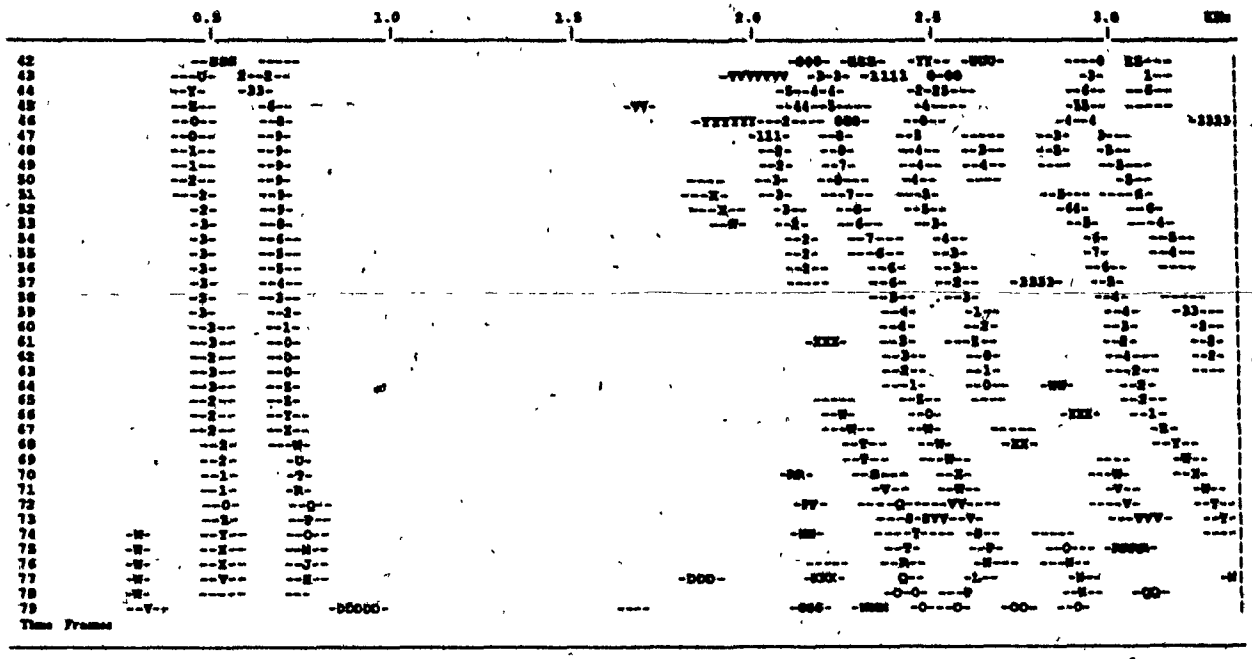


Fig. 3.4 Skeletonized pattern of Fig. 3.3 for letter "a".

3.4.4 Preprocessing

Preprocessing on skeletonized pattern is carried out to discard all isolated, weak, and scattered points in the pattern. Such points do not contribute to any meaningful assumptions either at grouping level or at hypothesis level. Preprocessing is carried out by applying an algorithm based on the strategy of *tracing continuity*.

The line tracing algorithm, LTA, retains properties like, collinearity, curvelinearity, continuity etc. present in the pattern. The non-accidental lines or significant lines in speech pattern are usually surrounded by lines which are less significant. The human vision system is capable of identifying such "biological camouflage" to a great extent. Similarly, when spectrogram experts read spectrograms, their vision system is capable of discarding non-significant (accidental) lines and to give emphasis on significant lines (non-accidental). Skeletonization and preprocessing are the two techniques which would give the machine similar capability as human to a certain extent in this context.

Thinning and preprocessing surface all significant and non-significant lines in the pattern and discard all scattered points. In terms of acoustic point of view, some of these lines are the formants. Some of the lines are *strong* and some are *weak*. The strength

and weakness of lines are measured in terms of energy and duration. Even though non-significant points are eliminated, non-significant lines are still present and keeping these lines is important in order to detect certain linguistic properties of speech.

Smooth lines are important perceptual structures themselves, as well as being needed for the subsequent detection of collinearity, curvilinearity, connectivity, and other such perceptual groupings.

3.4.5 The Line Tracing Algorithm (LTA_Algm)

The LTA accepts the skeletonized pattern and applies the following algorithm for smoothing. The skeletonized pattern is a binary image which contains only dark and white points. In Fig. 3.4, the points marked as "-" are also considered as white points.

The five-neighbors of point p are defined to be the 5 points adjacent to p, points, $n_0, n_1,\ldots,n_4$ as shown in Fig. 3.5. A continuous line, l, exists between points p_1 and p_n iff there exists a path $p_0p_1\ldots p_{i-1}p_i\ldots p_n$ such that p_{i-1} is a neighbor of p_i for $1<i\leq n$. A path between points p_i and p_{i+1} exists iff there exists at least one dark point among n_0-n_4 neighbours. If more than 1 dark point exists among n_0-n_4, then, the point n_j with the maximum energy is considered. If there exists more than one equally strong point, then the algorthim to find line, l, is recursively applied to find the line which is the longest from point n_j. The algorithm written in pascal like notation is given below:

N_0 (i-1,j)	N_1 (i-1,j+1)
N (i,j)	N_2 (i,j+1)
N_4 (i+1,j)	N_3 (i+1,j+1)

Fig. 3.5 Designations of the 5-neighbours of point N in a 3x2 window

```
line_tracing_algorithm (pattern:spectrogram; var vector;lines)

/ pattern is a binary image of the speech pattern /
/ vector will have all detected lines in the pattern /

begin
    set line counter, k = 0 ;
    for each row in pattern do
        begin
        for each column in pattern do
            begin
            set line_end = false;
            while not line_end do
                begin
                look for dark_point, p, in pattern;
                compute_neighbours [1], n, of point p;
                if n =0 then                    /end_of_line found/
                    if rule1 [2] then
                        begin
                        increment line_counter, k;
                        accept current_line as k;
                        set line_end = true;
                        end
                if n = 1 then                   /1 neighbour for p/
                    begin
                    accept point p for line, k;
                    set point p in pattern as white;
                    set new neighbour as point p;
                    continue tracing
                    end
                if n > 1 then                   /junction found/
                    begin
                    accept point p for line, k;
                    set point p in pattern as white;
                    set point p as strongest new neighbour;
                    continue tracing
                    end
                end_while
            end_do
        end_do
end
```

1: neighbours, n, is computed as,

$$n = (i\text{-}1, j) + (i\text{-}1, j+1) + (i, j+1) + (i+1, j+1) + (i+1, j)$$

where i and j points to the location of p in pattern.

2: rule1 = true , if

$k(p) > \psi_1$ and $k(h) > \psi_2$

where, k is the k^{th} line

p is the number of points in line k

h is the height of line, k

ψ_1 and ψ_2 were empirically determined constants.

The height of a line, h, is the number of time frames covered by the line. For example consider Fig. 3.6a to Fig. 3.6d which illustrate various ways lines may appear in pattern. Lines in Fig. 3.6a and 3.6b are not acceptable while those in Fig.6c and 3.6d are acceptable.

The idea of applying rule1 was not to eliminate redundant lines or accidental lines, but to eliminate those points which do not carry any acoustic information.

Fig. 3.7 shows the preprocessed pattern of Fig. 3.4. In Fig. 3.8, the line elements are replaced by the line number for conceptual reasons.

3.5 Perceptual Organization and Grouping of Lines

Skeletonized, preprocessed patterns are perceptual structures in themselves and can be used for further grouping. Preprocessing and line tracing can also be considered as a grouping process in biological vision where a series of discontinuous dots are grouped into oriented structures such as, lines and curves. According to Zucker[94], the process of grouping dots into oriented entities can be classified into two different catagories namely, Type I and Type II.

In Type I process, the lines and curves are highly accurate in position and curvature and variations may cause large apparent differences. Type II on the other hand are more tolerant of positional variations. They provide information about within-surface variations. Type II grouping process also provides a basis for a *first guess* about surfaces, such as where they are discontinous in orientation or reflectence and how they are curving [94]. First guesses are usally based on syntactic kinds of processing which may lead to a wrong guess. Therefore, these processes must have built-in safeguards which are provided by size/density constraints. It is extremely unlikely that collections of physical events will form images with randomly oriented structures, provided the events are of significant size and density [94].

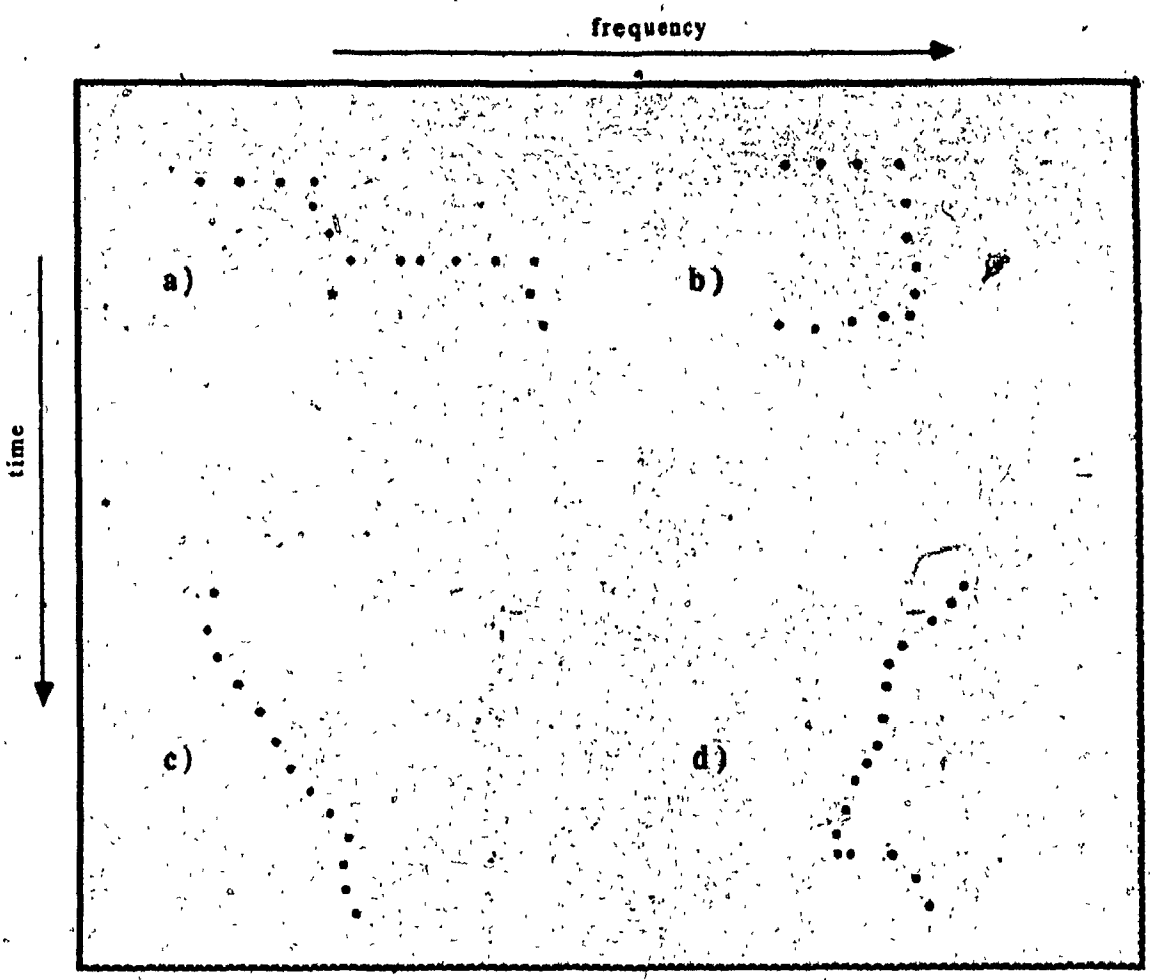


Fig. 3.6 Illustration of various Line formations

In the algorithm, LTA_Algm, when the number of neighbours of point p was greater than 1, the path chosen was the one with the strongest energy. This is an example of built-in safeguard to force the line or curve to take a natural course in its path tracing. In speech patterns, the lines formed are natural events and within these events the change should be minimal and hence, the energy difference between two successive points must be small. However, whenever there is a large difference, this marks the beginning of a new event.

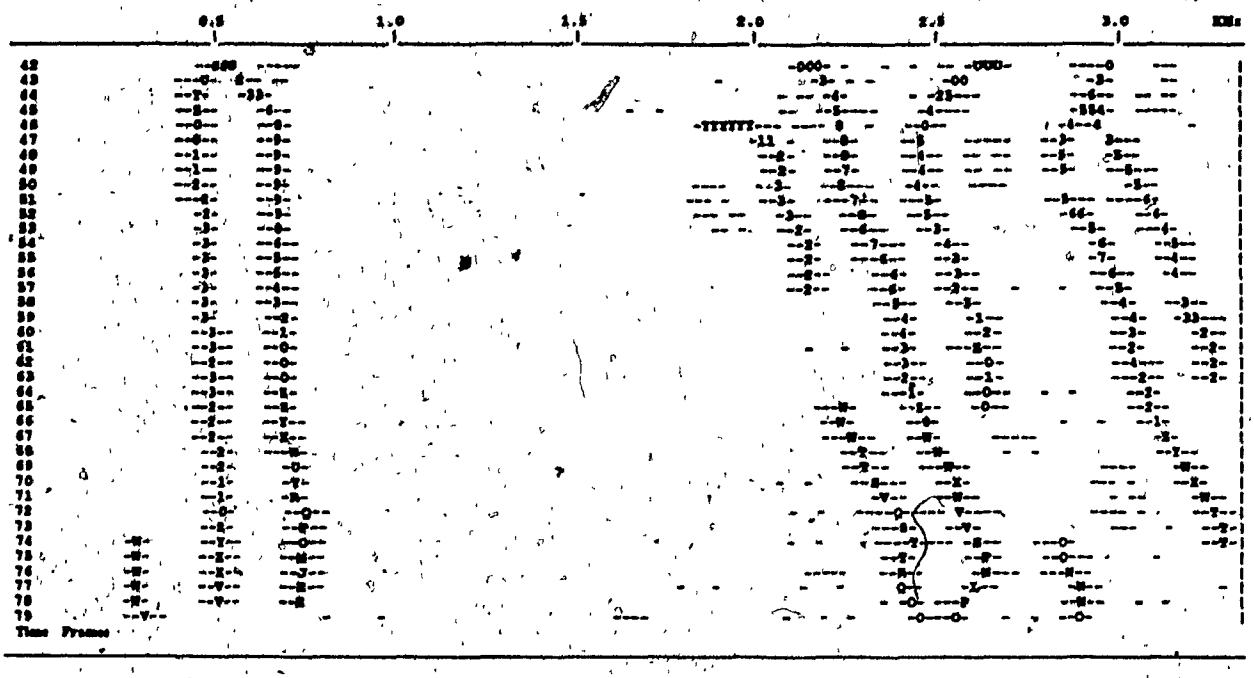


Fig. 3.7 Preprocessed pattern of Fig. 3.4 for the letter "a".

The separate existence of an object and the boundaries that define and divide it from its surroundings must be established before they can be considered for recognition [87]. Extensive studies on the phenomenology of perceptual organization showed that similarity, proximity, and common movement of discrete elements are strong determinants of perceptual grouping. The grouping strategy applied here takes into account the above phenomenon.

Description of grouped objects are carried out at various levels. At the lowest level description, level-0, we use a morphology in the time_frequency domain for characterizing phonetic events. Level-1 consists of describing each line in terms of its time evolution and frequency range. At level-2, temporal relations between line descriptions are considered. Among these relations, there are relations which are perceptually significant, like, *downward shift in the first formant region*. Details of various levels of description in the hierarchy are considered now.

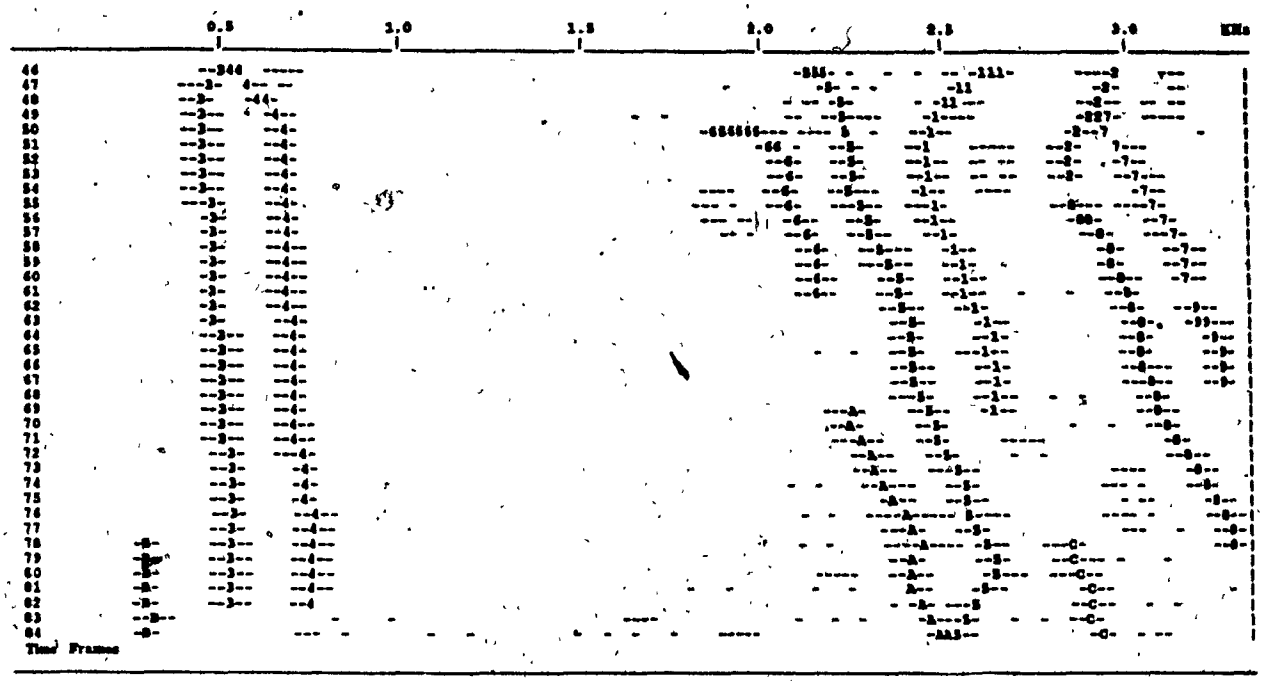


Fig. 3.8 Points on each line represented as line number for Fig. 3.7.

3.5.1 The Description hierarchy for Spectral lines

Level-0

The description hierarchy for spectral lines is based on acoustic properties that are known or are expected to be perceptually significant. The hierarchy follows an open taxonomy that can be expanded to incorporate new items and new classes. At the level-0 of the taxonomy, a spectral line is described by a sequence of vectors V_j of triplets,

$V_j = (t_{ji}, f_{ji}, e_{ji})$, $(j = 1..J; i = 1,...I)$ where,

t_{ji} is a time reference in centiseconds.

f_{ji} is a frequency value in Hz and

e_{ji} is an energy value in dB.

I is the length of line j

J is the maximum number of lines detected in pattern.

l_j is number of time frames (a time frame usually has a 10 msecs. duration and 200 speech samples) corresponding to the line duration. Each line sample is thus represented by its location, frequency, and energy. The line band width is not considered because it is in principle redundant and in practice difficult to estimate.

Level-0 grouping was based on the principle of Proximity, Colinearity, and Curvilinearity. The objective of level-0 grouping is to cluster collinear and curvilinear points. Since spectral lines do not always appear as straight lines, existing algorithms like Hough's transform[95] cannot be applied.

The information contained in vectors V_j can be further grouped into another level of the taxonomy by segmenting the spectral lines into segments of variable length as an acceptable approximation of their time evolutions.

Level-1

At level-1, spectral lines are described by morphology symbols, $x_k \in \Sigma_1$ and a sequence of attibutes. For example, the level-1 description a_k is expressed as follows:

$$a_k = x_k(ll_k\ tb_k\ te_k\ fb_k\ fe_k\ fM_k\ fm_k\ ea_k)$$

where,

$x_k \in \Sigma_1$ is a morphology symbol,

ll_k is the length of the line

tb_k is the beginning time of the segment described by a_k,

te_k is the ending time of the segment,

fb_k is the frequency at time beginning,

fe_k is the frequency at time end,

fM_k is the maximum frequency,

fm_k is the minimum frequency,

fa_k is the average frequency

ea_k is the average energy.

Symbols in Σ_1 gives a rough indication of the frequency location of the mid-point of the line and is defined as follows:

Σ_T: { LO: low; LA: low-average;
A: average; AH: average-high;
HI: high; VH: very-high; }

Notice that level-1 descriptions contain pointers k, tb_k, te_k that allows one to exactly pick-up the triplets of the level-0 descriptions corresponding to a_k.

Table 3.1 shows the level-1 description of the speech pattern shown in Fig. 3.8.

Table 3.1 Level-1 descriptions of speech pattern in Fig. 3.8

k	ll	tb	te	fb	fe	fm	fM	ea
1	29	41	65	2754	2700	2511	2754	8.1
2	10	41	49	3078	2916	2916	3078	8.3
3	37	42	78	513	540	459	540	8.1
4	39	42	78	540	783	540	783	7.8
5	47	41	81	2160	2727	2160	2727	7.9
6	18	46	57	1890	2187	1890	2187	8.1
7	12	45	56	3024	3240	3024	3240	8.8
8	25	51	74	2916	3375	2916	3375	8.3
9	7	58	63	3267	3348	3267	3348	8.4
A	18	65	81	2295	2565	2295	2565	6.8
B	10	74	83	297	297	297	324	7.1
C	15	74	83	2916	3213	2916	3213	5.6

3.5.2 Level-2: Identification and Description of Important Structural Features

As the grouping process continues into higher levels in the hierarchy, more structurally oriented and spatially organized properties are detected, described, and grouped.

Clustering of lines from several regions into different groups is the first stage of spatial organization at level-2. The clustering algorithm groups all lines into sets, VL, VM, and VH, where,

VL contains all lines in a_k, where, $x_k \in$ {LO, LA}

VM contains all lines in a_k , where, $x_k \in$ {A, AH}

VH contains all lines in a_k , where, $x_k \in$ {HI, VH}

This level of grouping is based on the principle of familiarity where the lines are grouped into three different frequency bands, namely, the low, medium, and high. All lines in each band which are seen being together are grouped together. Acoustic reasoning behind this grouping is that, changes in place-of-articulation is usually visible in these three different bands.

Detection of Anchor Line

Using set VL, we now detect one of the most important property of a specific line, £, called the "anchor line" . Most of the subsequent level grouping and descriptions will have a direct influence of line, £. Line £ is the "most significant" line in the VL range which could be considered as the "first formant". Since formant tracking or any direct hypothesis based on formants are "not" addressed in this work, we refer to this line as the anchor line.

£ is detected by applying an algorithm which uses certain perception rules. The algorithm to detect £ is given below:

Algorithm: Detect Anchor-Line (DAL_Algm)

Rule-0: find the strongest line, S_i and the next strongest line, S_j , such that,

$S_i, S_j \in VL$

Rule-1: find the longest line , L_i , such that, the energy of

line $L_i \geq Oe$, where, Oe is the overall average energy of all lines in a_k

Rule-2: set $£ = S_i$

Rule-3: if [length(L_i) > α_1 * length(S_i)] then $£ = L_i$

Rule-4: if {abs[strength(S_j) - Strength(S_i)] < α_2} and [$fa(S_j) < fa(S_i)$]

then $£ = S_j$

- Rule-0 through Rule-4 are applied sequentially.

- Strength of line is the ea_k component in a_k
- Length of line l, is the ll_k component in a_k
- α_1 and α_2 are empirically determined constants set equal to 2
- recall that fa represents the average frequency

Elaboration of Rules

- Rule-0 picks up the strongest line, S_i, as £ initially.
- Rule-1 overrides Rule-0 if there exists a sufficiently strong line for which,

$$|L_i| \geq \alpha_1 * |S_i|$$

- Rule-2 tests whether there exists equally 2 strong lines, S_i and S_j, such that, $|S_i - S_j| < \alpha_2$ and if S_j lie below S_i in frequency, then S_j, the weaker among the two, is the anchor line. This rule is significant in cases where there exist many strong lines in low frequency region as in the case of back vowels.

Locational property of the anchor line is very significant and is used extensively later on for the description and hypothesis of other properties.

3.5.3 Level-3 Inter-line Relationships

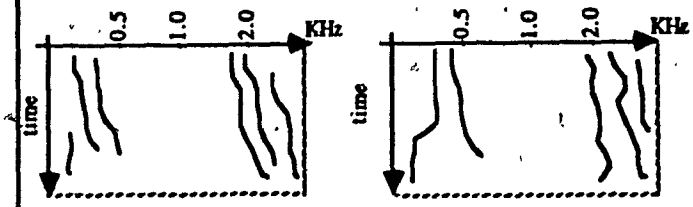
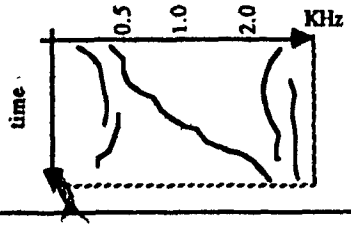
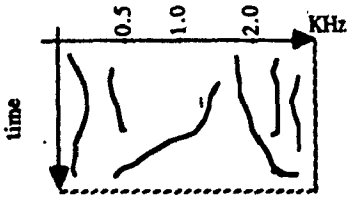
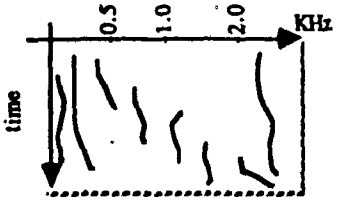
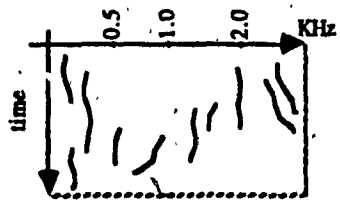
Descriptions at this level are more perceptually oriented. Level-3 descriptions refer to level-2 descriptions and they represent "temporal relations" within level-2. The relations are of type:

Γ_1: $\Re_m \{ y_{m1}, y_{m2}, \ldots, y_{mk} \}$ where, $m \leq k$

$\Re_m$ is a relation symbol belongs to an alphabet Σ_2 and $y_{m1}, y_{m2}, \ldots, y_{mk}$ are descriptors like a_k belonging to Σ_1. The alphabet Σ_2 for $\Re_m$ is defined in Table 3.2.

The properties in Γ_1 represent the spatial relationship among lines in specific localities. Each property is detected from a different context and represents different spatial relationship. For example, the property STCR, represents a certain property among many lines in a certain frequency range, while the property, FDN, represents the

Table 3.2 Properties and descriptions of perceptual components

Components	comments	stylized patterns of properties
FDN	anchor line shifts downward along frequency	
ASND	lines grow upward along frequency from low-to-mid or from mid-to-high frequency bands	
DSND	lines grow downward along frequency from mid-to-low or from high-to-mid frequency bands	
STCR	Ascending lines are not continuous and appear as stair cases	
STCL	Descending lines are not continuous and appear as stair cases	

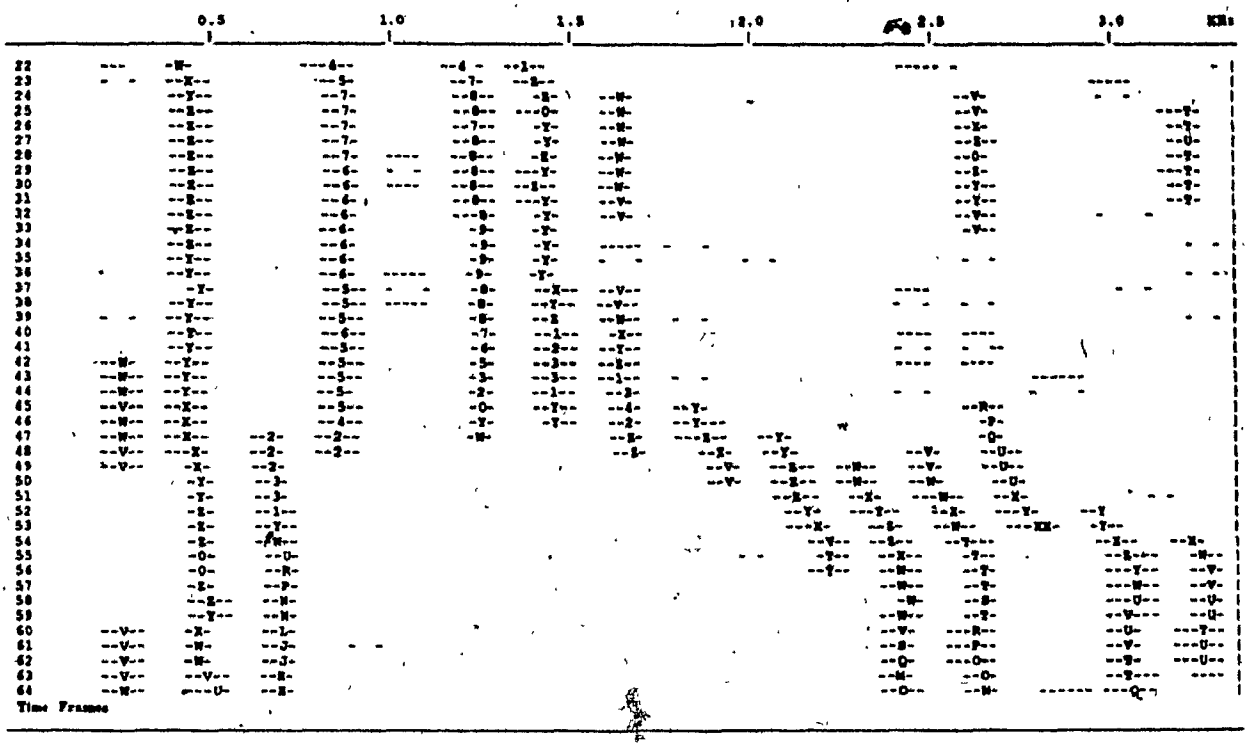


Fig. 3.9 Pattern of a diphthong /ai/. The transition for /I/ starting at frame # 47.

presence of a particular relationship among two lines.

Most of the properties in Γ are detected in low and medium bands using sets VL and VM. Temporal changes in patterns are greatly affected in these two regions whenever changes in place-of-articulation occur. Fig. 3.9 shows a typical pattern of diphthong, */ai/*, where the change of evolution of lines in time are clearly visible when the place-of-articulation changes from central to front. A detailed discussion of the detection of each component in Γ_1 is considered now.

Rules for Detecting Relation symbols, $\Re_m$ in Γ_1

Rule-1: $\Re_m = \text{FDN}$

The relation $\Re_m$ has parameters, the number of which varies depending on the type of property detected. In the case of FDN, the relation involves two lines, i.e.

$$\Gamma_1 : \Re_m \{y_{mi}, y_{mj}\}$$

where, y_{mi} is a line in VL and y_{mj} is the anchor line, £. Relation FDN is detected when the following condition hold for the attributes of lines y_{mi} and y_{mj}:

$$\{\exists [i \in VL] \mid fm_i \prec fm_j;\ ea_i > ea_j;\ i \text{ succeeds } j \text{ in time}\} \cup$$

$$\{\exists [\text{shift in } j] \mid ea_{jshifted} > ea_{jstable}\ ;\ fm_{jshifted} < fm_{jstable}\}$$

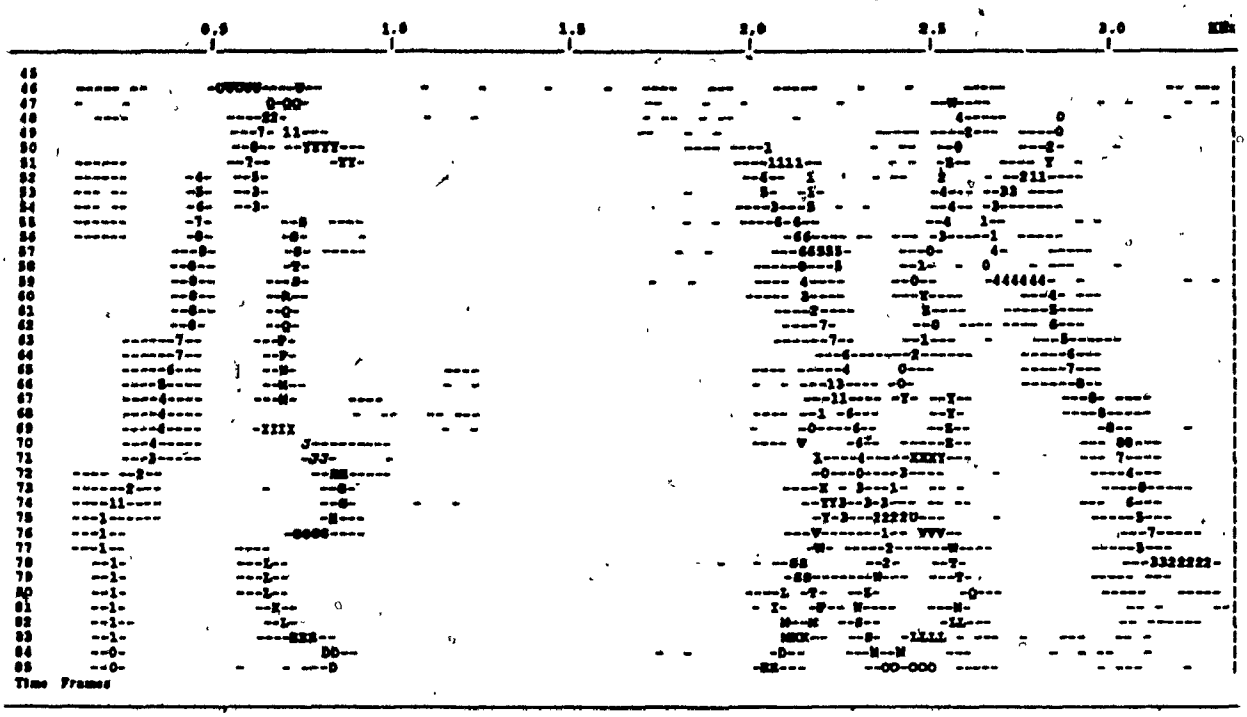


Fig. 3.10a Example of a "follow-down" property. Case(1).

The first condition of the disjunction states that if there is a strong line following the anchor line in time and if there is a shift towards low frequency, then FDN property exists. Fig. 3.10a shows examples of these situations.

The second condition states that, if the anchor line, j, has a sharp shift in frequency towards low region and the energy of the shifted region is sufficiently strong compared to that of the stable part (region before the shift occured), then there is evidence that FDN property exists. Fig. 3.10b shows an example of such a situation.

From the acoustic point of view, the FDN property reflects a situation where a transition in place-of-articulation has occurred and the transition is greatly affected in the low band region.

Rule 2: $\Re_m$ = ASND, DSND

For ASND and DSND,

$\Gamma_1 : \Re_m \{ y_{mi} \}$ where, $ym_i \in \{ VL, VM \}$.

ASND and DSND properties are detected based on whether the time evolution of the line j is slanting to the left or to the right. As it has been observed in speech patterns, spectral lines are usually curvilinear. For example, in Fig. 3.11, line marked as "J" is neither ascending or descending while the one marked "A" can be considered as descending.

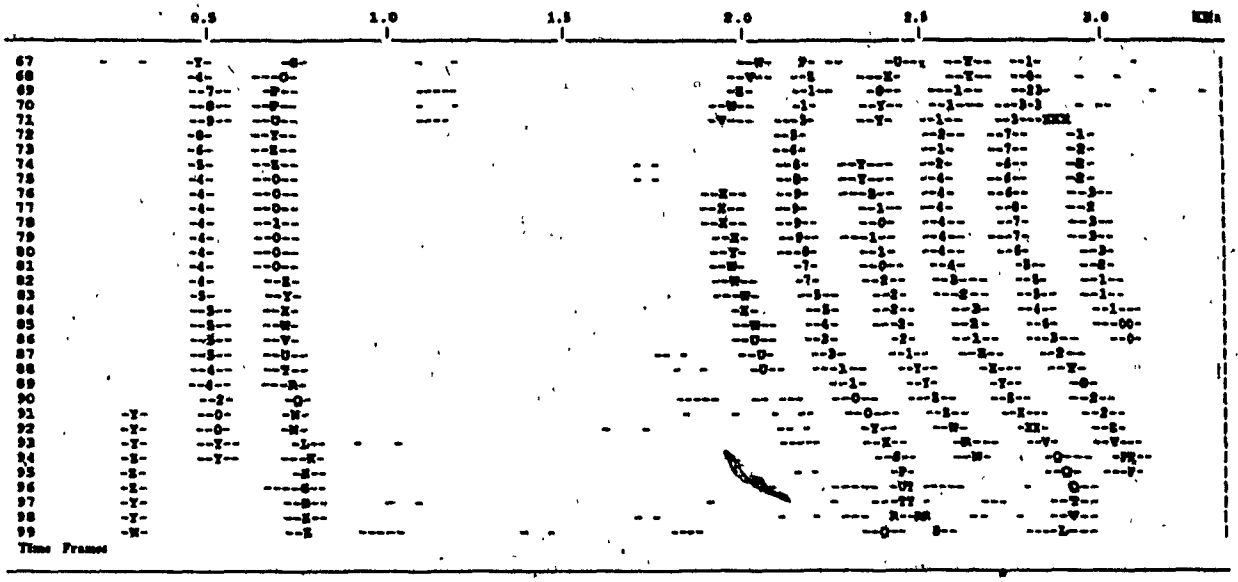


Fig. 3.10b Example of a "follow-down" (FDN) property: Case(2)

These properties are detected by computing the slope at each point with respect to the initial point. The slope S_j of line j, is calculated as follows:

$$S_j = \sum_{i=n+1}^{N} \frac{[1 - \frac{f_{jn}}{f_{ji}}]}{N}$$

where,

N = total number of points in line j

n = initial point in line j

f_{jn} = frequency of the n^{th} point of line j

f_{ji} = frequency of i^{th} point of line j

also, $n = 1$ initially and $(n+1) \leq i \leq N$

Once the slope, S , has been calculated, the decision of whether a line is slanting to the left or to the right is determined by computing the truth of the following conditions:

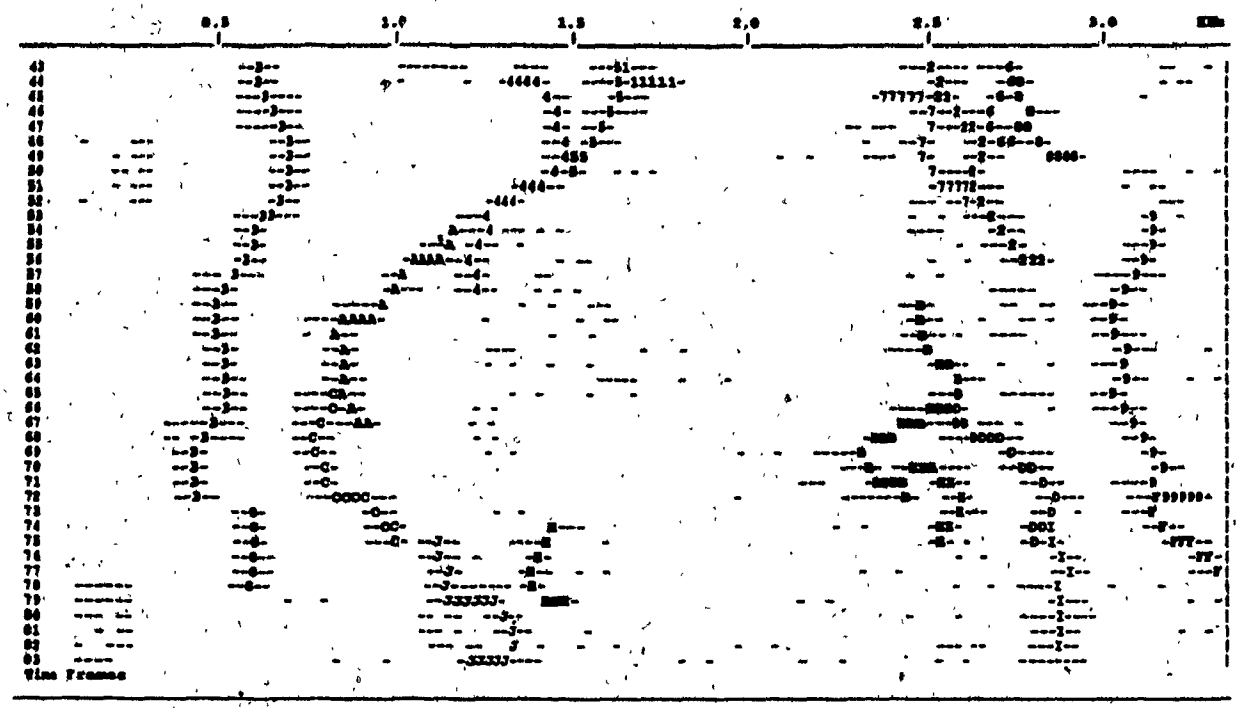


Fig. 3.11 Example of non-descending (line "J") and non-ascending (line "B") lines.

$$\text{ASND} := S_j \geq \theta_1 \text{ and}$$

$$\text{DSND} := S_j \leq \theta_2$$

where, θ_1 and θ_2 were determined empirically and set at 0.8 and -0.8 respectively.

ASND and DSND properties are evident in low and mid band regions if there is a transition in place-of-articulation. The originating and terminating locations in time and frequency of these properties are highly speaker dependent, but the properties are important cues for certain class of speech sound. For example, a line with ASND mid-to-high frequency range is a cue for a central-to-front vocalic transition for the diphthong /ai/.

There are also cases when both ASND and DSND property coexist in the same acoustic segment. An example of such a situation is shown in Fig. 3.12

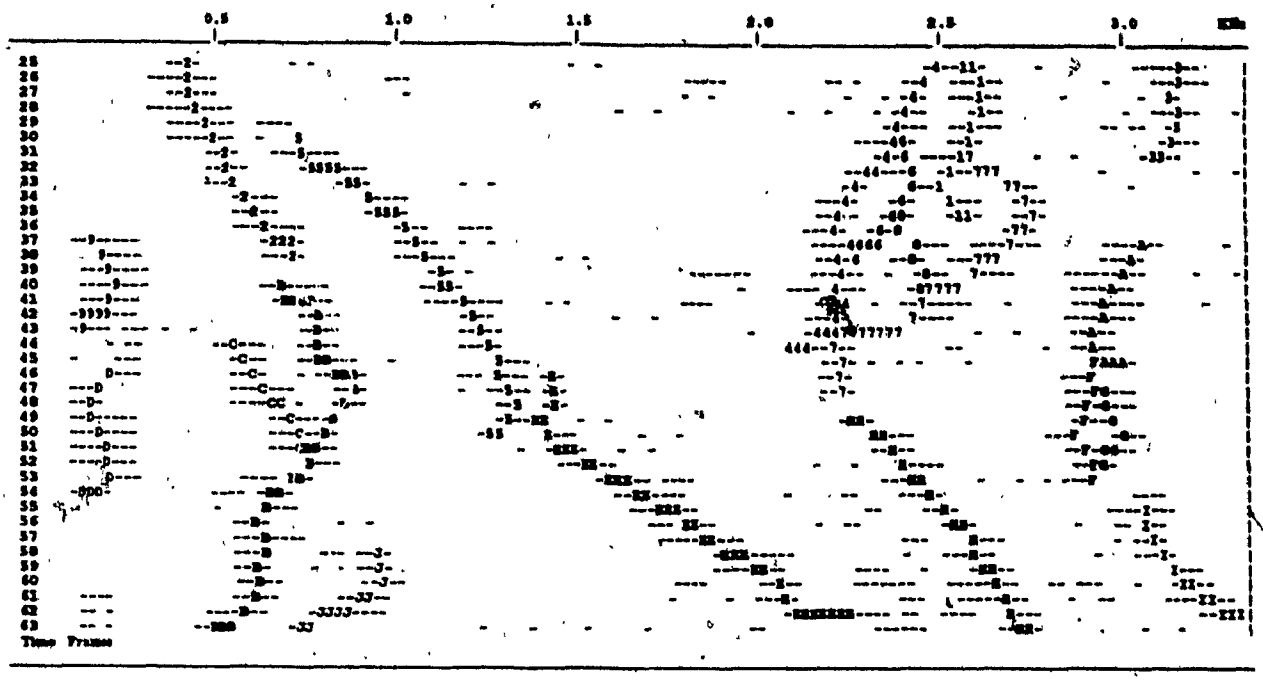


Fig. 3.12 A situation where both ASND and DSND properties coexists. (Line marked "B" is descending and line "E" is ascending)

Rule 3: $\Re_m$ = STCL, STCR

These two properties convey the same acoustic-phonetic information as in the case of ASND and DSND except that, the physical phenomenon of slanting to the left or to the right, is present in a camouflaged manner. To elaborate on this point, in the case of ASND or DSND, the property is clearly represented by smooth curved lines, where, in some cases, the slanting is not smooth but is exhibited through several disjoint, quasi-parallel lines which are shifted in frequency as well as in time. These shift in frequency and time make these properties look like stair-cases. The stair-case property is characteristic of certain speakers. For some speakers the transitions are smooth while for others the transitions are rather rough. Fig. 3.13 shows examples of both STCL and STCR property.

For STCL and STCR, the relation function is defined as,

$$\Gamma_1 : \Re_m \{(y_{mi}, y_{m1}), (y_{mi}, y_{m2}), \ldots, (y_{mi}, y_{mj}), \ldots, (y_{mi}, y_{mJ})\}$$

where,

$$y_{mi} \in VM; \quad y_{mj} \in VL, VM, VH; j \neq i; j \leq k$$

The $\Re_m$ properties are detected with respect to a certain line, the top-line, i,

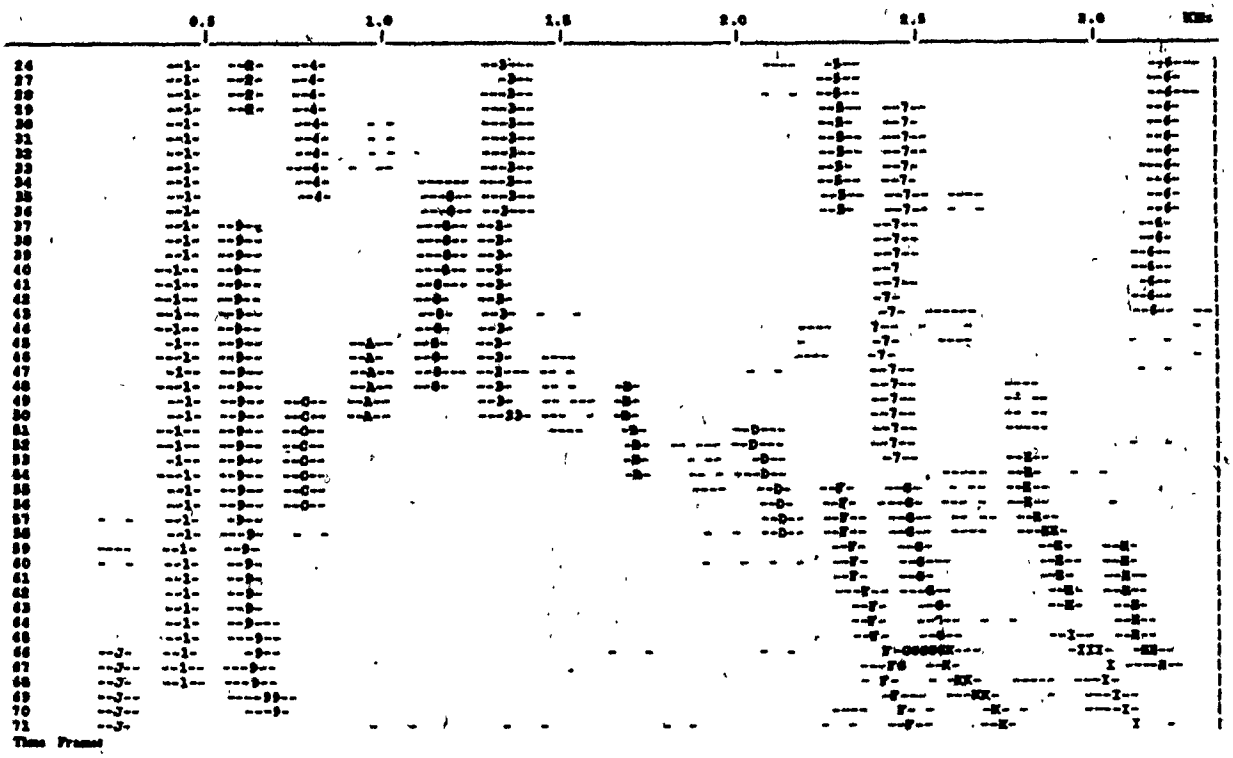


Fig. 3.13 An illustration of STCR and STCL properties. Line marked as "3" is the top-line. Lines: "8", "A", and "J" belongs to STCL. Lines: "B", "D", "F", and "K" belongs to STCR.

which is the strongest(in energy), shortest, earliest (in time), and must lie in the mid-band range. Each line, $j \in k$, are checked against, i, for the property of STCL and STCR based on the following algorithm:

Algorithm: Detect STCL-STCR (STCR_STCL_Algm)

condition-1:

If $[\exists i \mid i \in k;\ i \in VM;\ te_i \leq te_j;\ ea_i \geq ea_j \mid]$

then condition = TRUE

where, $j = 1 \ldots L, j \neq i$, L is the maximum number of lines in VM.

condition-2:

a. compute set, W_l such that W_l contains all lines on the left of i with the property,

$$W_{lj} \in \{VL, VM\}; tb_{W(lj)} < tb_i; fa_{W(lj)} < fa_i$$

b. compute set, W_r such that W_r contains all lines on the right of i with the property,

$$W_{rj} \in \{VM, VH\}; tb_{W(rj)} < tb_i; fa_{W(rj)} < fa_i$$

c. sort W_l based on frequency, high-to-low

d. sort W_r based on frequency, low-to-high

e. for $n = 1$ to L do
    begin
        STCL exists between i and n, if, $tb_n > tb_i$
        $i = n$
    end
for $n = 1$ to R do
    begin
        STCR exists between i and n, if, $tb_n > tb_i$
        $i = n$
    end

where, L is the total number of lines in W_l and R is the total number of lines in W_r.

Comments

- condition-2 is activated only if condition-1 is true.
- if L = 0, then STCL does not exist.
- if R = 0, then STCR does not exist.

For all of the above described properties, FDN, ASND, DSND, STCL, and STCR, an attribute called the "measure of strength" is also attached with the property relations. The measure varies depending on the properties and evaluation of measure is as follows.

3.5.4 Measure association for detected properties

For each property symbol in Γ, a so called "measure of strength" is attached at the time of detection of the property. This measure is always a quantity between 0.0 and 1.0 inclusive. The measures are determined differently for each property symbol and is defined as follows:-

FDN

Measure for FDN is 1.0, if the property is present and 0.0 if not present.

ASND and DSND

For these two property symbols, the measure is the slope computed for each

line. Since the more ascending or descending a line is, the slope value will be greater which actually reflects the strength of the property.

STCR and STCL

The measure for these properties are computed with respect to the *top-line* and all other lines on the left and right of the *top-line*. The distance in frequency between the *top-line* and the k^{th} line under consideration is manipulated in such a way as to give a measure between 0.0 and 1.0. The line which is closest to the *top-line* has a higher measure than a line which is farther away. For example, a line VM will have higher measure than a line VL. These measures reflect the spatial correspondence between the top-line and the rest of the lines.

3.6 Chapter Summary

The major points addressed in this chapter are:

- Speech spectrogram is a valuable tool in ASR research and it carries significant acoustic and phonetic informations.
- Expert spectrogram readers apply biological vision techniques while reading spectrograms and uses prior linguistic knowledge to identify them.
- Based on Gestalts theory on Spatial correspondence and Perceptual Organization, a new technique has been proposed to give machine biological vision capability and to use this capability for interpreting speech spectrograms.
- Speech spectrograms were treated as patterns and as an early grouping process, the patterns were skeletonized, preprocessed, and described into level-0, level-1, level-2, and level-3 in the hierarchy of description.

Chapter 4

Low-level Segmentation and Stochastic Learning Using Perceptual Components

In the last chapter we discussed the details of low-level and higher level grouping processes in the hierarchy.

In this chapter we will see how perceptually grouped elements can be used for low level segmentation of acoustic segments. Low level segmentation or internal segmentation is a process for detecting quasi-stable intervals within a larger acoustic segment. Each interval will then contain a piece of phonetic information which can be detected more reliably. Final phonetic hypotheses can be generated by concatenating the pieces of phonetic information detected.

We will also present in this chapter a novel approach to recognize phonetic information in speech segments based on Hidden Markov Models (HMM).

4.1 Internal Segmentation of Speech Patterns

The speech segments that can be processed with the techniques proposed in this chapter are the sonarant ones. These acoustic segments may contain more than one vowel. For example, in the case of diphthongs, there is a change in place-of-articulation like, back-to-central or central-to-front, within the acoustic segment. Therefore, the vocalic properties before the transition and after the transition are different. There are also cases where the transitions occur within the same region. For example, in the case of phoneme, /æl/, the change occurs within the low frequency region, { LO, LA}.

Transitions of this nature cannot always be detected by classical formant tracking algorithms because formant trackers are not perfect. Therefore, predicting vocalic properties based on formant transitions may be not reliable. A typical case where a formant tracker may fail is shown in Fig. 4.1.

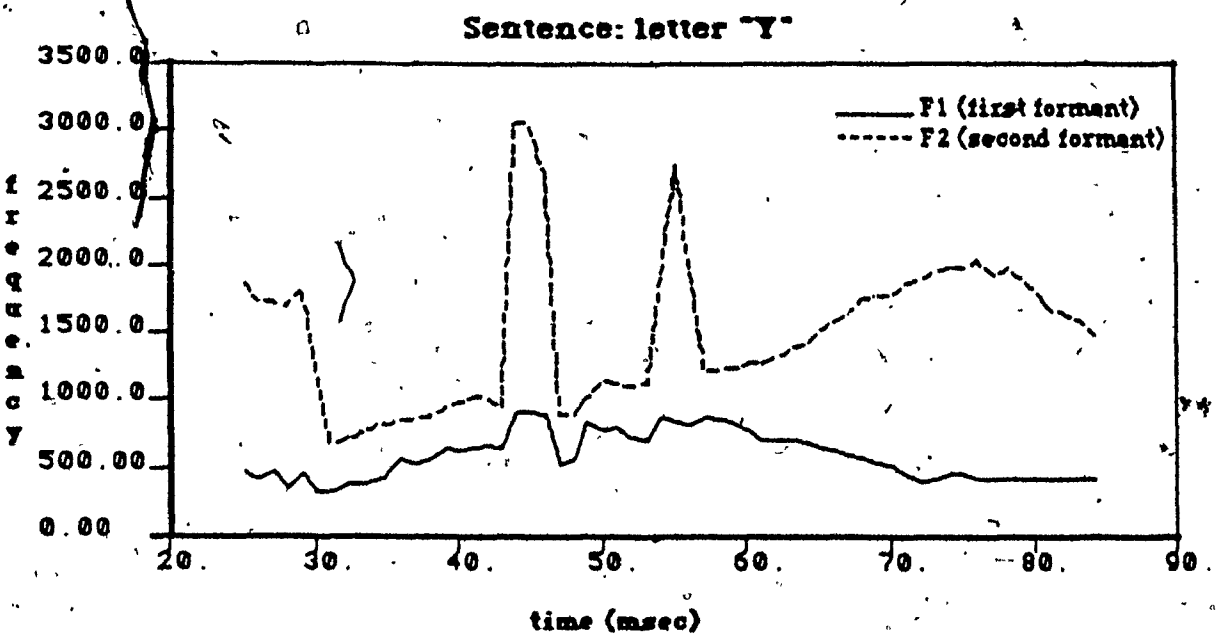


Fig. 4.1 Plots of 1st and 2nd formants of an utterance of a diphthong obtained using classical formant tracking technique.

In order to capture intra-segmental variations, an internal segmentation algorithm is applied on the acoustic segments. The internal segmentation algorithm would sub-segment the original acoustic segment whenever there is an unexpected behavioral change within the pattern.

Internal segmentation is carried out on two different dimensions of the pattern, the energy and frequency. In both cases, energy and frequency of lowband lines are coded using a set of symbols. Segmentation is then performed on the coded string of symbols.

4.2 Frequency Based Coding and Segmentation

The first step towards internal segmentation is the coding of lines in the low band region in such a way that the coded string would represent the continuity behavior of the strongest point with respect to time and frequency. The coding algorithm uses a set of symbols belonging to the alphabet β_1 :

$$\beta_1 = \{ G, S, J, D \}$$

The symbols represent the type of frequency changes occuring within the band. The meaning of each symbol of the alphabet:

G: Gap
S: Stable
J: Jump
D: Drop

We now consider the algorithm for coding. As mentioned earlier, only the lines in low frequency band, VL, are considered for the coding. The coding algorithm basically looks for the strongest point, the anchor point, for each time frame in the acoustic segment.

Algorithm: Frequency Based Coding (FBC_Algm)

```
begin
    tf = tbeg                                  /tf is the first time frame/
    find_panc(panc,tf)                         /find the anchor point/
    code_panc(panc,tf)                         /code the anchor point/
    tf = tf + 1                                /look into next frame/
    while ( tf ≤ tend ) do                     /find anchor point and code/
        find_panc(panc,tf)                     /it for the rest of time frames/
        code_panc(panc,tf).
        tf = tf + 1
    end_while
end
```

Comments on the FBC_Algm

- *find_panc*: finds the strongest energy point, p_{anc}, in the low band region for the time frame t_f
- *code_panc*: the algorithm for coding uses the following rules:
  - Rule-1: *if* ($t_f = t_{beg}$) *then code* = S /initial point/
    Rule-2: *if* ($p_{anc} = nil$) *then code* = G /no points in frame/
    Rule-3: $\Delta_f = \{ frequency[p_{anc}(t_f)] - frequency[P_{anc}(t_{f-1})] \}$
    if $\Delta_f < \theta f$

then code = D

else if $\Delta_f > \theta_1$

then code = J

else code = S

- If there are no frequency points in the time frame t_f, then there is a gap and the code is "G".
- Code "S" corresponds to stability in the low band with respect to the previous anchor point, if the frequency of the current anchor point is within $\pm \ \theta_1$.
- If the difference, Δ_f, between frames t_f and t_{f-1} is less than θ_1 then there is a downward shift in the anchor line which is coded as "D" and if the difference is greater, then the shift is upwards and the code is "J". Symbols "J" and "D" hints the possiblity of change in the place-of-articulation.

The symbol, describing frame t_f, indicates the stability of the pattern between frames t_f and t_{f-1}. The constant, θ_1 can be adjusted so that the shift up or down can be controlled. The value of θ_1 has been set as the value of one spectral point which is equal to 37Hz. The coded-string is a representation of the perceptual group, where the grouping was based on continuity. The string of β_1 is kept in a vector F_{code} which will be used later for segmentation.

In certain cases, the string generated after coding may contain situations where oscillation occur. For example, consider the two strings a) and b) generated by the algorithm:

a) ....SSSSSDJSSSSS....

b) ....SSSSSJDSSSSS....

In both cases, there is a very short oscillation which causes a stable region to drop-jump and stabiliz or jump-drop and then stabiliz. Since sonorant regions should be stable for at least 30 to 40 msec., such oscillation can be ignored and hence, the

the string is rewritten according to the following rules:

...SDJS... ⇒ ...SSSS...

...SJDS... ⇒ ...SSSS...

However, if the oscillation repeats, i.e. $(JD)^+$ or $(DJ)^+$, then the region is heavily unstable and can be considered as non-sonorant. Fig. 4.2 shows a pattern segment with a coding string after rewriting.

4.2.1 Frequency Based Internal Segmentation Algorithm (FBS_Algm)

The coded and rewritten string can be used for internal segmentation. The strategy for segmentation is based on a simple rule that a bound is set whenever, any of the symbols, "G", "J", or "D" followed by the symbol "S" marks the end of previous segment and the time of the next symbol is the beginning of a new interval. The algorithm for frequency based internal segmentation is given below:

Algorithm: Frequency Based Segmentation (FBS_Algm)

```
set internal segment counter, j = 0;
repeat
    while {[F_code(i) = G ] and [ p ≤ t_end ]} do        /skip all gaps/
        increment, p, the time frame counter;
        increment, g, the gap counter;
    end_while

    test for property NSPH/NSPT;

    reset gap counter to zero;
    increment internal segment counter, j;
    starting time of internal segment, FBS_j = p;

    while {[F_code(i) = S] and [ p ≤ t_end] } do          /count all S/
        increment, p, the time frame counter;
    end_while;

    ending time of internal segment, FBS_j = p -1;
until p > t_end;
```

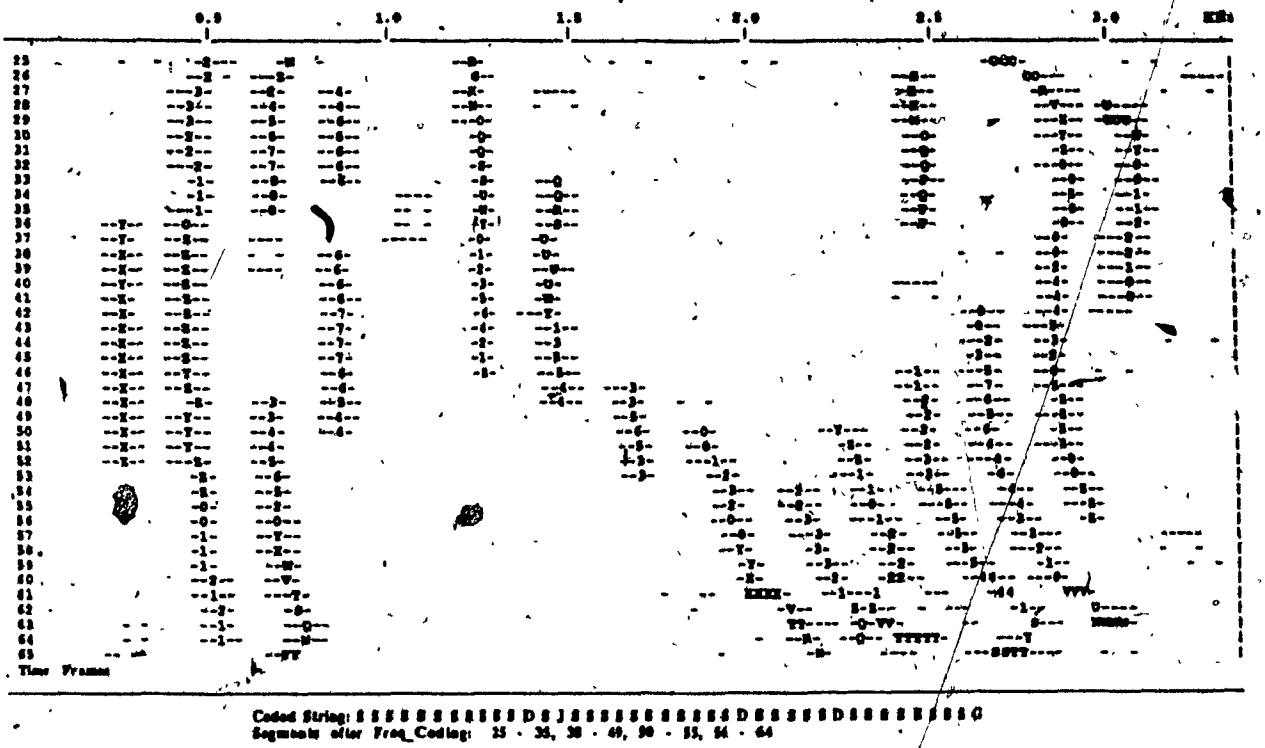


Fig. 4.2 Example of pattern showing the string after frequency coding.

Comments on Algorithm FBS_Algm

- Two significant acoustic properties belong to the set; Γ_2 = {NSPH (non-spectral head) , NSPT (non-spectral tail)} are detected at this point. If there are several consecutive gap symbols at the beginning of an acoustic segment then NSPH is present; if the gap symbols are present at the end, then NSPT property is present. The rules to confirm NSPT and NSPH property are: look for the location of the gap (head or tail), and use the length of gap given by the gap counter, g. The measure signifying the strength of the property is based on the length of the gap.
- $i = 1..l..I$, where l is the time frame index of F_{code} and I is the total number of symbols in F_{code}.

The frequency based internal segmentation process will correctly segment the pattern whenever a "sharp" change in place-of-articulation occurs. The term "sharp" is relative and can be adjusted by changing the value of θ_1. Fig. 4.3 shows a sample pattern marked with frequency based internal segments.

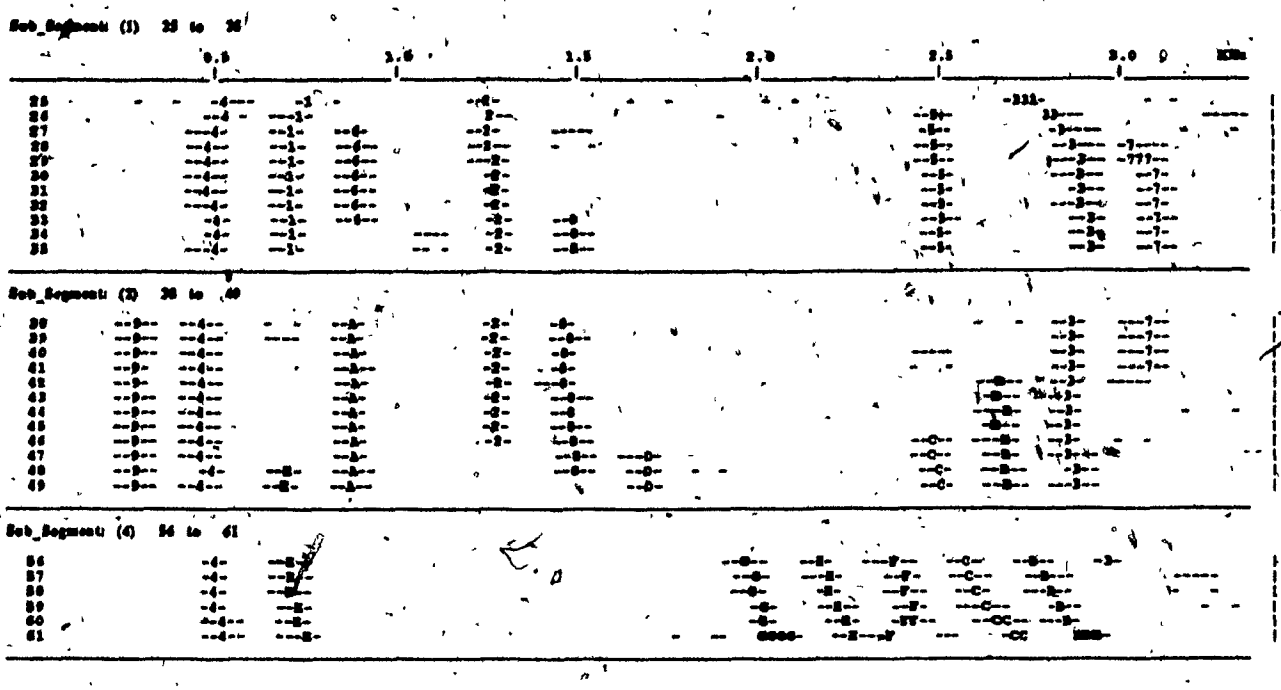


Fig. 4.3 Internal segments after coding for pattern in Fig. 4.2

There are some cases where there are no changes in the frequency region but a transition in place-of-articulation does exist. A typical example is the word "zero" where the transition from /e/ to /o/ is rather smooth for some speakers. Since no drop or jump may be present in the anchor line, the FBS_Algm will not generate internal segments. However, capturing the transition is important for the correct recognition of phonemes. An example of such a situation is shown in Fig. 4.4. An energy based internal segementation process is applied in order to capture such transitions.

4.3 Energy Based Coding and Segmentation

Energy based coding and segmentation is carried out only on internal segments which were already generated by frequency based segmentation. The internal segments which are input to EBS-Algm are stable with respect to frequency and time but may not be stable with respect to energy and time.

A smooth rise and fall in the energy of the anchor line is a typical property of any single vocalic segment. However, if there is a change in place-of-articulation for which the frequency evolution in time is smooth, there will be more than one rise-and-fall in the energy contour of the anchor line. This property can be observed in Fig. 4.4.

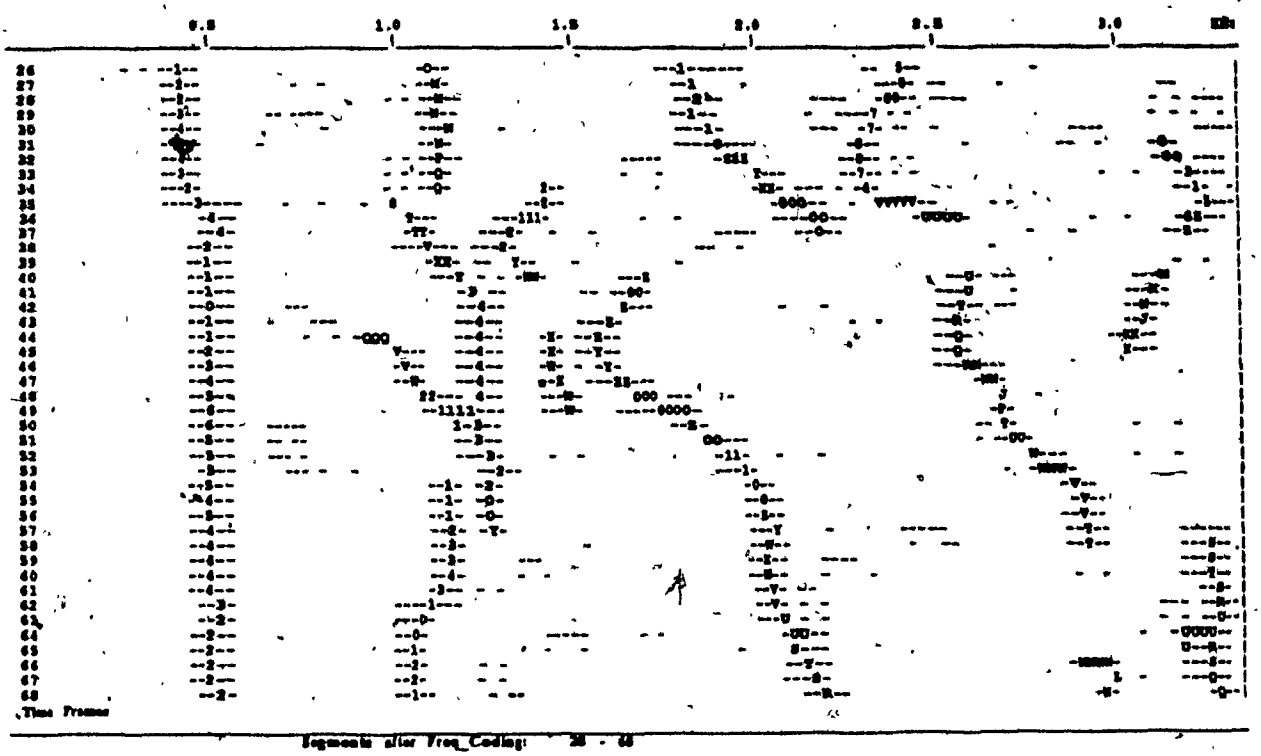


Fig. 4.4 Example where FBS algorithm is not sufficient for internal segmentation. The above pattern corresponds to digit "zero" and /o/ starts at ≈53.

Energy based coding uses symbols similar to those used in frequency based coding except that, there is no gap symbol "G" in the alphabet. The alphabet for energy coding, β_2 is:

$$\beta_2 = \{S,J,D\}$$

where, the symbols represent energy changes occuring within the already detected frequency based internal segment. The contextual meaning of each is the same as in frequency code.

Algorithm: Energy Based Coding (EBC_Algm)

1. $E_{code}[start\_point]$ = "S" /start with stable code/
2. *for i = start_point + 1 to end_point do*

   begin

   if { *energy* $[F_{code[i]}]$ > *energy* $[F_{code[i-1]}]$ }

   then $E_{code[i]} = J$

   else if {*energy* $[F_{code[i]}]$ < *energy* $[F_{code[i-1]}]$ }

   then $E_{code[i]} = D$

else $E_{code[i]} = S$

end

Comments on EBC_Algm

- E_{code} is a vector which will contain the energy coded string.
- *Start_point* is first point in the frequency coded internal segment vector, F_{code}.
- *End_point* is the last point in the internal segement, F_{code}.

The energy coded string can be segemented again to form more internal segments. The algorithm for energy based segmentation looks for the behaviour "rise-fall-rise" in the energy coded string. Note that, if there is no rise-fall-rise property in the frequency coded internal segment, then the internal segment would be left as it is. The energy based internal segmentation algorithm (EBSAlgm) is given as,

4.3.1 Energy Based Internal Segmentation Algorithm (EBS_Algm)

For each internal segment generated by FBS_Algm, do

begin

for i = starting_point to ending_point do

if sequence J followed by D exists

then $EBS_j = i$ /*i* marks a segment/

end_do

end

Comments on EBS_Algm

- *starting_point* is the beginning of a frequency based internal segment.
- *ending_point* is the end of frequency based internal segment.
- the algorithm looks for the rise in energy after a drop (*J* followed by *D*).
- If above property is found, then enter the time into vector EBS which contains all internal segments generated by EBS_Algm.

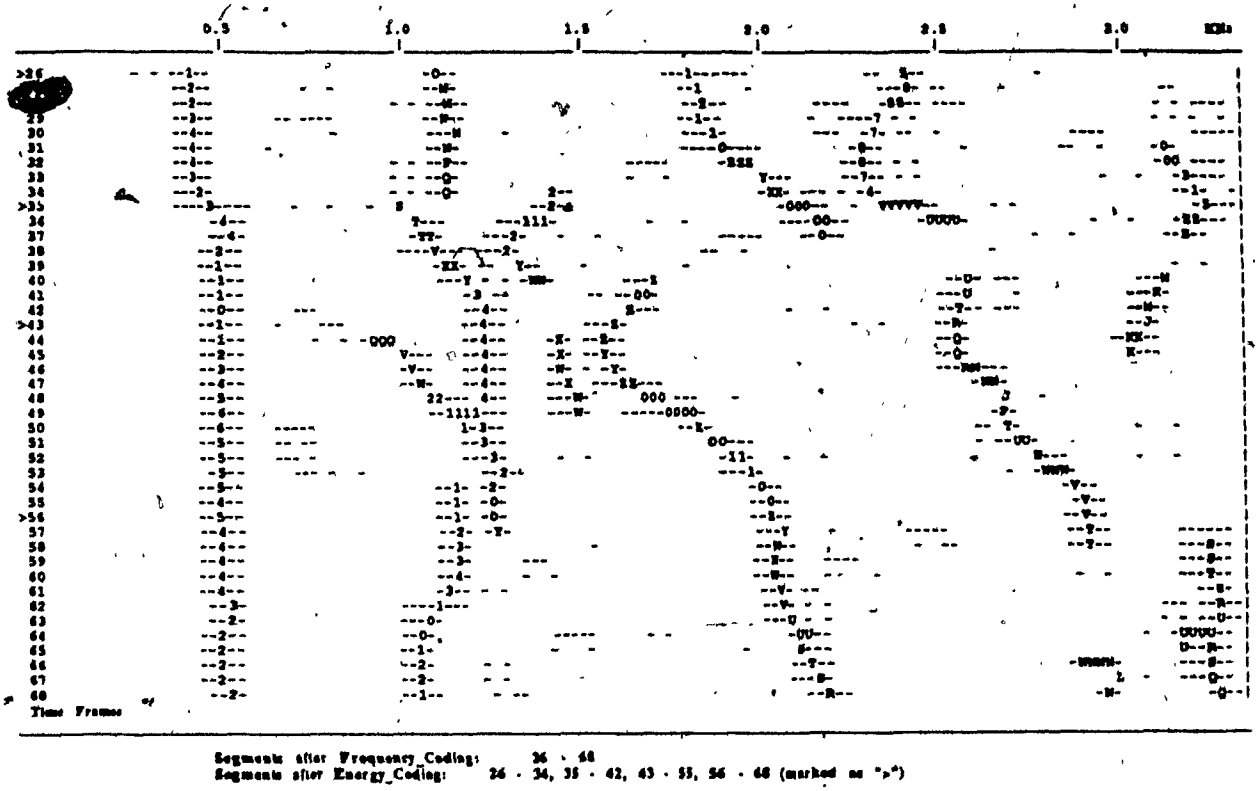


Fig. 4.5 Internal segments obtained after applying FBS and EBS algorithms for pattern in Fig. 4.4

Fig 4.5 shows the final set of internal segments obtained after applying FBS_Algm and EBS_Algm. The segments are kept in a vector S, where,

S_i = *(starting frame number, the ending frame number)* of the i^{th} internal segment.

Each internal segment, generated by FBS_Algm or EBS_Algm, reflects a region in the pattern which is stable with respect to frequency and energy. If we associate phonetic meaning to these segments, we assume that the place-of-articulation of each segment is unique. In the coding and segmentation process, attempts were made in order to obtain "over segmentation" rather than "under segmentation". This would allow one to reduce the probability of losing variations occuring in the pattern.

In the case of over-segmentation, more than one consecutive segment may be detected for the same vowel. Such segments can be, by later processing, combined and considered to be a single vocalic. Under-segmentation may cause significant variations undetected in the pattern and would cause both properties to be missed and possible misrecognition when hypothesis are generated.

We have discussed various levels of grouping and description which are perceptually significant. Internal segmentation based on frequency and energy were also very important perceptual groups. Knowingly or unknowingly, these are some of the basic steps followed by spectrogram readers. The grouping and description are made possible by biological vision phenomena, and phonetic and linguistic knowledge are

applied to these described elements for identification. An expert spectrogram reader would look for significant spatial variations that are present in different localities and would then predict his/her conviction.

4.4 Phonetic Level Identification of Internal Segments

Since speech signals are highly variable, their properties will also vary to a great extent. Absence of properties, shift in the values of properties, accidental appearance and disappearance of properties, properties which are camouflaged, are just a few of the variations which are observed in speech patterns. Because of these reasons, pure rule based or even pure stochastic model based recognition systems are unable to produce consistent robust results for difficult vocabularies pronounced by many speakers. It can be argued also that, expert speech spectrogram readers can still manage to identify spectrograms in the presence of the above mentioned problems. "Act according to situation" and "proceed based on what is available" is the usual strategy applied by spectrogram readers.

Internal segments which were obtained are perceptual groups which contain accidental as well as non-accidental properties. Instead of detecting and eliminating the accidental ones, the segment is considered as a whole and used for learning as a whole under different phonetic contexts. Since we are considering only vocalic segments and the segments themselves are stable in nature, the important phonetic properties one has to derive from these segments are the place-of-articulation (PA) of the vowels. An alphabet for PA is:

$$\Gamma_3 = \{ \text{VB: back vowel), (VC: central vowel), (VF: front vowel)} \}$$

4.4.1 Markov Models in Speech Recognition

Perceptually grouped and described components can be used for phonetic level recognition of speech sounds. Even though these components are not sufficient to generate robust hypotheses, they can be used in conjunction with other evidence or they can be used for deriving other evidence for final hypotheses generation.

The basic theory of Markov Chains has been known to the scientific world as early as 80 years ago. However it is only during the past two decades that Markov Models are used in Speech Processing area. The main reason for this is the lack of

methods available for optimizing the parameters of the Markov models to match the observed signal patterns.

Markov models can be applied to any real world problem which produces a sequence of observable "symbols" [96]. The symbols could be discrete or continuous as in the case of speech signals. If a signal model, which explains and characterizes the occurrence of the observed symbols can be built, then this model can later be used to identify other sequences of similar observations.

Speech signals change with time and a model for a speech signal must address temporal variations also. However, speech signal has the property that, within a "short time" (20 msecs - 30 msecs) period the signal is stable and, therefore, the short -time segment can be effectively modeled by a simple linear time-invariant system. Time varying nature can then later be accommodated by concatenating the smaller "stationary segments". The stationary segments (short-time segments) do not have any pre-defined time duration. In speech, as well as in many real world problems, the signal properties within short-time periods may remain steady and change rapidly or smoothly to exhibit another type or same type of properties as time changes to next short-time period.

If the following 3 problems can be correctly answered, then HMM's can be effectively used in speech processing problems. They are:-

1. how to identify the distinctively behaving short-time periods in Speech Signals.
2. how to characterize the sequentially evolving properties of Speech signals for short-time periods.
3. what common short-time model to be chosen for each short-time periods.

The above mentioned problems have been solved and several ASR systems have been developed in the past [52]. In most of the existing ASR systems which use Markov Models, the output of the Markov Chain are strings whose symbols belong to a finite set of alphabet and are generated sequentially over the time domain.

Usually left-to-right models as shown in Fig. 4.6 are used in speech recognition applications. In this model, a low numbered state always precedes a high numbered state and this inherently imposes a temporal order to the HMM, since lower numbered states account for observations occuring prior to those with high numbered states. The basic mechanism of a HMM for speech is as follows:

- There are a finite number of states, N, in the model and within each

state the signal possess some distinctive and qualitative property.

- At each clock time, t, a new state is entered based on a transition probability distribution which depends on previous state.
- After each transition, an observation output symbol is generated according to a probability distribution which depends on the current state.

The compact notation to represent an HMM is given as,

$$\lambda = (A, B, \Pi)$$

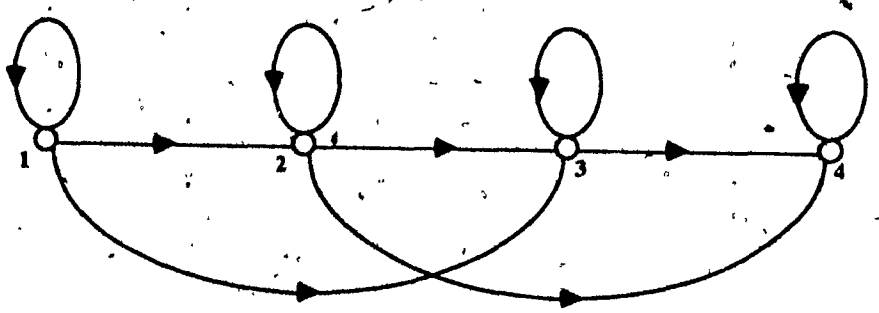


Fig 4.6 Model of a left-to-right HMM

The specification of such an HMM involves choice of the number of states, N, and the number of discrete symbols, M, and the specification of the three probability densities A, B, and Π. For a model with N states and M symbols, Π is an initial state probability vector $(Nx1)$, A is a transition probability matrix (NxN), and B is a state output symbol probability matrix (NxM). These matrices completely specify the model. For example, $\Pi(i)$ is the probability of starting at state i, $a(i,j)$ is the probability of moving from state i to state j, and $b(i,o)$ is the probability of observing output symbol o from state i.

In order to use the model for any application, there are 3 problems to be solved.

1. Evaluation Problem

   "Given a model and a sequence of observations, how to score or evaluate the model". In other words, given the observation symbol,

$O = O1,O2,....,OT$ and the model, $\lambda = (A,B,\Pi)$, how to compute $Pr(O|\lambda)$, the probability of the observation symbol. Mathematical solution to this problems can be found in literature [96], [77], 52]. One popular approach is the forward-backward algorithm [96].

2. Estimation Problem

   Given the observation sequence, $O' = O1,O2,....,OT$, how to choose a state sequence, $I = i1,i2,...iT$ which is optimal in some meaningful way. A formal technique for finding such an optimal path is by using the Viterbi algorithm [96].

3. Optimization Problem

   How to adjust the model parameters, $\lambda = (A,B,\Pi)$, to maximize $Pr(O|\lambda)$. This is called the training sequence where we train the model which allows us to optimally adapt the model parameters to observed training data. A formal technique called Baum-Welsch re-estimation formulas [96] is used to solve this problem.

Returning to the example of an isolated word recognizer using HMM, let us consider the following. Assume we have a vocabulary of V words to be recognized. We also have a training set of L tokens of each word and an independent testing set. In order to design a recognition system, the following tasks are involved:

- Build an HMM for each word in the vocabulary. Observations from the training set of L tokens can be used to estimate the optimum parameter for each word. Thus we have a model λ^v for the v^{th} element of the vocabulary, $1 \leq v \leq V$.
- For each unknown word in the test set, which produces the observation sequence, $O = O1,O2,....,OT$, and for each word model λ^v, calculate $Pv = Pr(O|\lambda^v)$ using the solutions to the 3 problems of HMM described previously.
- The recognized word has the model whose probability is the highest, i.e. $V^* = argmax_{1 \leq v \leq V}[Pv]$

This is a classical approach of using a HMM for speech recognition. Once the 3 basic problems, evaluation, estimation, and optimization are solved, the performance of a speech recognizer will heavily depend on the parameters which were used. Different

kinds of parameters are used and usually they are generated sequentially over the time domain.

We propose a HMM in which symbols are substituted by perceptually grouped elements from the hierarchy of descriptions. The components which are considered are the spectral lines which were sequentially generated over "frequency domain" in level-0 descriptions.

4.5 A Continuous Parameter and Frequency Domain Based Markov Model (CPMM)

Instead of generating symbols of a finite alphabet, the parameters used are "spatial relations" and "locational informations" among spectral lines which were originally generated over frequency domain. Each spectral line is represented by continuous distribution of parameters. Switching from time domain to frequency domain drastically reduces the number of states on the Markov chain and the use of continuous parameters completely eliminates quantization error.

So far, HMMs have been successfully used for modelling time varying processes representing spoken words and sentences. The advantages of such models when applied to ASR systems have been reported in [52], [97].

Recently, Kopec [98] has applied Markov models to formant frequency estimation showing new possible application of such a paradigm in aspects of speech processing other than time varying processes. In this work we propose an application of HMM in another aspect in ASR, such as, characterization of the place-of-articulation of vowels for a large population of speakers and a large variety of contexts.

Internal segments obtained after FBS_Algm and EBS_Algm are effectively used for the learning and identification of place-of-articulation. For the learning and recognition of vocalic regions, Hidden Markov Models (HMM) are used with a novel approach.

Hidden Markov Models have been used in the past in the area of speech recognition with great success. HMM's are used in this context as a tool to learn and identify the place-of-articulation of sonorant regions.

The performance of an ASR system based on HMM's depends more on the "type" and "quality" of the parameters used rather than the model itself. The parameters used in our model are perceptually grouped elements which are continuous and extracted in frequency domain and, hence, we call the model "Continuous Parameter and Frequency domain Based Model" (CPMM).

4.5.1 The Continuous Parameter Markov Model (CPMM)

A CPMM is a Markov Model in which transition produce real vectors of parameters. The probability $P(s1,s2)$ defines the probability of choosing the transitions from $s1$ to $s2$ when state $s1$ is reached. $Q(s1,s2,v)$ is the probability that the vector $v = p1,p2,....,pn$ is produced when the transition t from $s1$ to $s2$ is chosen. The collection of probability distribution of the parameters describes a transition. A transition t is then described by the following matrix:

$$M_t = \begin{bmatrix} m_1 & \sigma_1 \\ m_2 & \sigma_2 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ m_n & \sigma_v \end{bmatrix}$$

Figs. 4.7a, 4.7b, and 4.7c shows the CPMMs corresponding Back, Central, and Front vowel models.

The input to the CPMM is a string of vectors of the form, $x = v1,v2,....,vm$

The vectors are obtained from internal segments of pattern as described in section 4.2. The internal segments obtained are sonorant portions of the signal which are quasi-stationary. From these quasi-stationary segments, the vector x is obtained from some of the morphological properties which were already extracted at various levels in the hierarchy. In the vector sequence, the first set corresponds to the anchor line. The remaining lines in the pattern are sorted by frequency and the corresponding vectors are added to x. Each vector has two components, which are defined as follows:

$$V_1 = p_{11}, p_{12}$$

$$V_i = p_{i1}, p_{i2}$$

$$V_m = p_{m1} \cdot p_{m2}$$

where,

p_{11} = frequency of anchor line

p_{12} = energy of anchor line

$p_{i1} = f_i - p_{11}$

$p_{i2} = e_i - p_{11}$

f_i = frequency of the i^{th} sorted line in the pattern

e_i = energy of of the i^{th} sorted line in the pattern

In order to obtain normalization, difference between frequency of anchor line and frequency of the i^{th} line (p_{i1}) and difference between energy of anchor line and energy of i^{th} line (p_{i2}) are considered.

The process that is modelled can generate spectral lines and their energies. The model contains non-accidental lines (e.g. formants) as well as accidental lines (e.g. noise). Frequency and energy distributions are associated to each transition in the model. The model is conceived in such a way that variances of the distribution are kept small so that each distribution really represent variation due to inter-speaker differences of the parameters of a line having specific structural properties.

For both learning and recognition, the Forward-Backward algorithm was used. During the recognition process, the FB-algorithm is used in order to compute the probability, $p = (x/M_j)$, where x is the input string of vectors and M_j is a CPMM. The string x is assigned to the i^{th} class if,

$$\{p(x/M_i)\} = \max_{\{M_j\}} \{p(x/M_j)\}$$

The a-priori probability computed by CPMMs are confirmed by 3 rules. These rules will correct any trivial error which may have made by the CPMM. For example, one of the observed errors is made by front vowel CPMM when the following stituation occured:

If{ [frequency of anchor line is high (≈ 700 hZ)] and
[no lines present in the mid band region] and
[strong or very strong lines in high band region] };

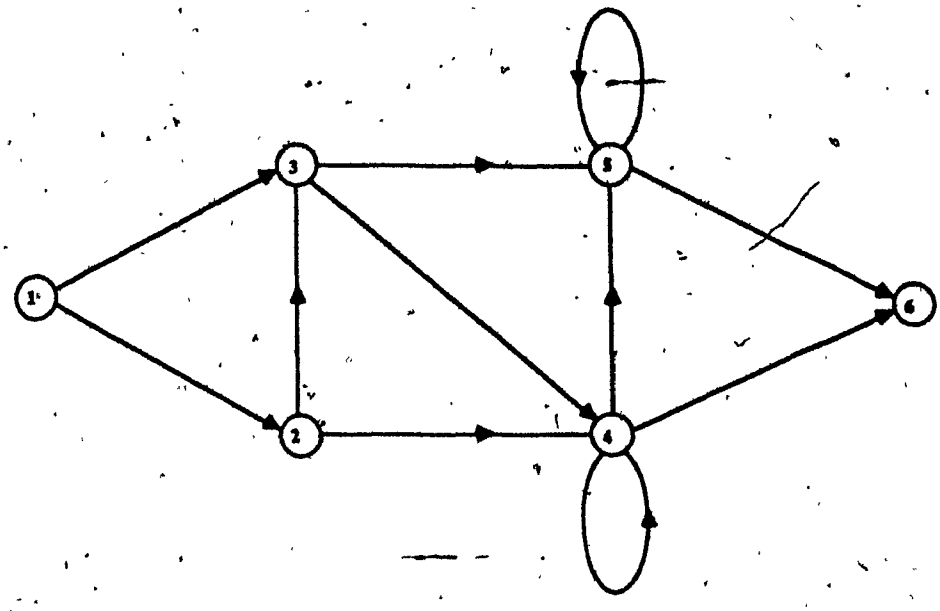


Fig. 4.7a CPMM for "Back Vowel" (transition parameters are shown in Table 4.1)

Transition		Probability of Transition	Parameter # 1		Parameter # 2	
from state	to state		Mean (M)	Variance (σ)	Mean (M)	Variance (σ)
1	2	0.550	570.090	1851.513	8.809	0.191
1	3	0.450	528.140	2384.136	8.438	0.369
2	3	0.459	206.060	29406.381	-0.350	0.020
2	4	0.541	253.393	3334.415	-0.257	0.140
3	4	0.878	393.276	19150.404	-0.168	0.134
3	5	0.122	1883.019	4183.322	-1.531	0.024
4	4	0.323	880.184	67645.172	-1.941	0.833
4	5	0.576	1819.642	153428.703	-2.176	0.517
4	6	0.101	2701.823	15439.756	-1.966	0.061
5	5	0.443	2218.530	96298.328	-2.202	0.708
5	6	0.557	2737.890	8447.664	-1.614	0.809

Table 4.1 Transition Parameters for "Back Vowel" (Ref. Fig. 4.7a)

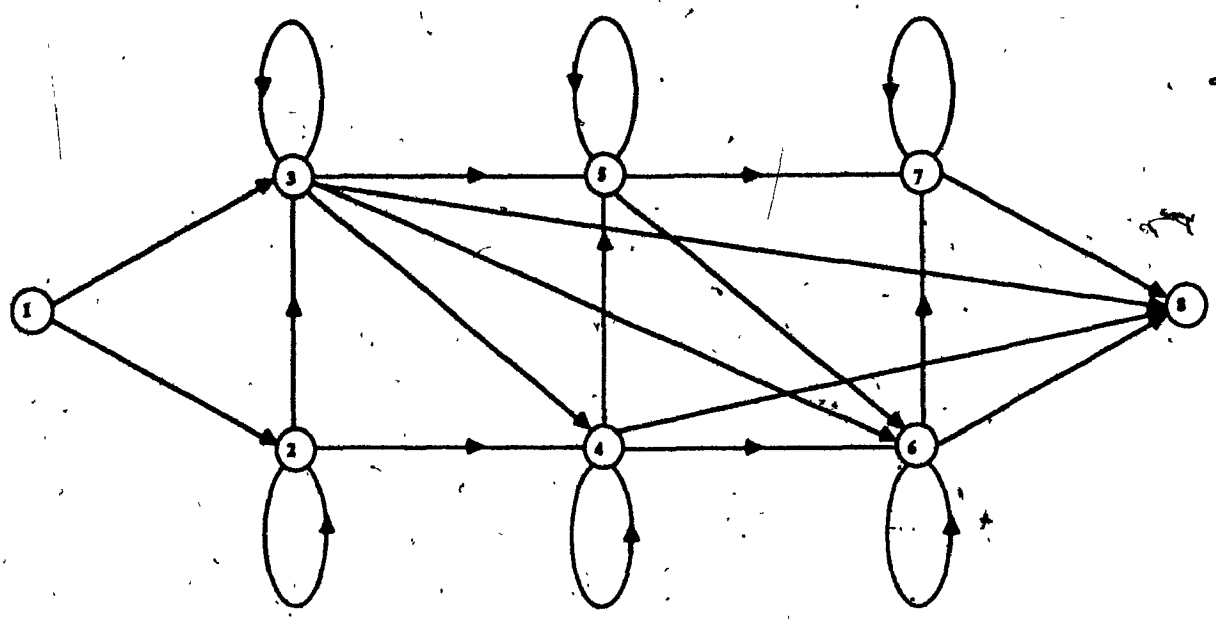


Fig. 4.7b CPMM for "Vowel Central" (transition parameters are given in Table 4.2)

Transition		Probability of Transition	Parameter # 1		Parameter # 2	
from state	to state		Mean (M)	Variance (σ^2)	Mean (M)	Variance (σ^2)
1	3	0.221	777.974	7244.006	8.576	0.577
1	2	0.779	783.254	7430.651	8.587	0.574
2	4	0.354	392.941	507329.094	-0.655	0.447
2	3	0.326	289.525	482435.750	-0.693	0.460
2	2	0.320	183.919	495948.813	-0.752	0.469
3	8	0.000	962.596	878899.875	-0.591	0.431
3	6	0.095	632.267	569753.625	-0.576	0.416
3	5	0.146	393.876	470285.313	-0.629	0.453
3	4	0.728	429.573	518125.344	-0.637	0.443
3	3	0.031	308.035	499673.813	-0.681	0.462
4	8	0.000	1705.290	547970.875	-0.573	0.445
4	6	0.358	922.124	426738.938	-0.405	0.342
4	5	0.532	681.043	330763.125	-0.388	0.357
4	4	0.111	760.141	401895.469	-0.402	0.358
5	7	0.572	1014.282	419429.719	-0.399	0.338
5	6	0.363	960.179	421785.219	-0.398	0.342
5	5	0.065	725.972	329779.094	-0.379	0.351
6	8	0.150	1901.754	441204.188	-0.575	0.455
6	7	0.818	1033.238	419284.938	-0.398	0.337
6	6	0.033	1000.254	415977.719	-0.393	0.338
7	8	0.828	2081.247	274714.938	-0.617	0.481
7	7	0.172	1222.221	396869.563	-0.390	0.362

Table 4.2 Transition Parameters for "Vowel Central" (Ref. Fig. 4.7b)

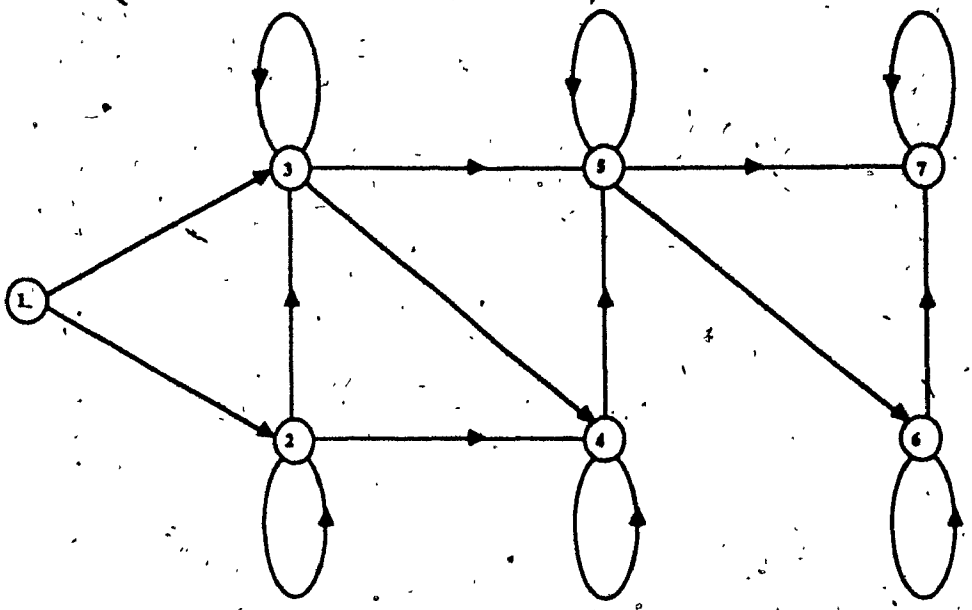


Fig. 4.7c CPMM for "Vowel Front" (transition parameters are given in Table 4.3)

Transition		Probability of Transition	Parameter # 1		Parameter # 2	
from state	to state		Mean (M)	Variance (σ)	Mean (M)	Variance (σ)
1	3	0.438	458.790	8935.368	8.033	0.386
1	2	0.562	522.003	4019.573	8.488	0.293
2	4	0.509	311.666	6104.020	-1.726	0.091
2	3	0.327	228.448	1803.066	-1.261	0.500
2	2	0.164	-136.760	14552.347	-0.812	0.223
3	5	0.488	1777.933	57545.125	-0.016	0.498
3	4	0.297	639.481	275764.000	-1.222	0.878
3	3	0.215	444.053	6388.213	-1.855	0.191
4	5	0.704	1731.968	39209.133	-0.340	0.631
4	4	0.296	1502.419	14644.381	-0.467	0.858
5	7	0.357	2320.153	89535.836	-0.159	0.536
5	6	0.342	2052.133	4344.315	0.054	0.561
5	5	0.302	1865.370	8709.803	-0.251	0.722
6	6	0.217	2157.887	2999.774	0.072	0.858
6	7	0.783	2323.118	14175.450	0.132	0.380
7	7	1.000	2665.696	28239.914	-0.014	0.662

Table 4.3 Transition Parameters for "Vowel Front" (Ref. Fig. 4.7c)

then the vowel central CPMM produced the highest probability, even though it was not a central vowel. The reason for this was that the model for central vowel has learned that the vowel is central whenever the frequency of the anchor line is very high even if there are no lines in the middle band, which is also a necessary requirement for a vowel to be central.

Simple rules based on spatial information about acoustic properties of speech can be used at this point without causing any computational overhead. The rules used to confirm vowels are given below:

Rule_Confirm_Front_Vowel

IF [Pr(front vowel is highest)]
AND [(number of lines in high band ≤ 0)
OR (average energy of lines in high band << average in mid band)]
THEN find_next_best_candidate /not front vowel/
ELSE vowel confirmed

Rule_Confirm_Central_Vowel

IF [Pr(central vowel is highest)]
AND (number of lines in mid band ≤ 0)
THEN find_next_best_candidate /not central vowel/
ELSE vowel confirmed.

Rule_Confirm_Back_Vowel

IF [Pr(back vowel is highest)]
AND [number of lines high band >> number of lines in low band]
AND [energy of high band >> energy of low band]
THEN find_next_best-candidate /not back vowel/
ELSE vowel confirmed

Corrections are seldom needed when the candidate selected by the CPMM is back vowel. Find_next_best_candidate is a function which selects the next candidate with highest probability and applies the corresponding rule recursively until either all candidates are tested or vowel is confirmed. If all candidates are tested and none was

confirmed, then an error is flagged.

4.5.2 Frame Analysis Using CPMM

When a-priori probabilities computed by different CPMMs were close, such that $p < \delta$ (p is the absolute difference between the two closest probabilities and δ is a constant set at value 1.0), then a technique called frame analysis is performed, on the internal segment under consideration, for vowel confirmation. In frame analysis process, each time frame is analyzed and labelled using a symbol of the alphabet, §, where,

$$\S = \{ VB, VC, VF, ER \},$$

{VB = back vowel; VC = central vowel; VF = front vowel; ER = error}

by the CPMM. In the previous model, parameters for CPMM were generated by considering the whole internal segment in which the strings were the relative difference between frequencies of anchor line and all other lines as well as difference between energies of anchor line and all other lines.

In frame analysis, instead of having an anchor line, the anchor point is first detected from the low band region. The parameters for the CPMM are computed as before by considering the anchor point and the rest of the points in the time frame instead of lines. Therefore, the description of parameters in the input string, x, has been changed as follows:

p_{11} = frequency of anchor point

p_{12} = energy of anchor point

$p_{i1} = f_i - p_{11}$

$p_{i2} = e_i - p_{11}$

f_i = frequency of the i^{th} sorted point in the pattern

e_i = energy of of the i^{th} sorted point in the pattern

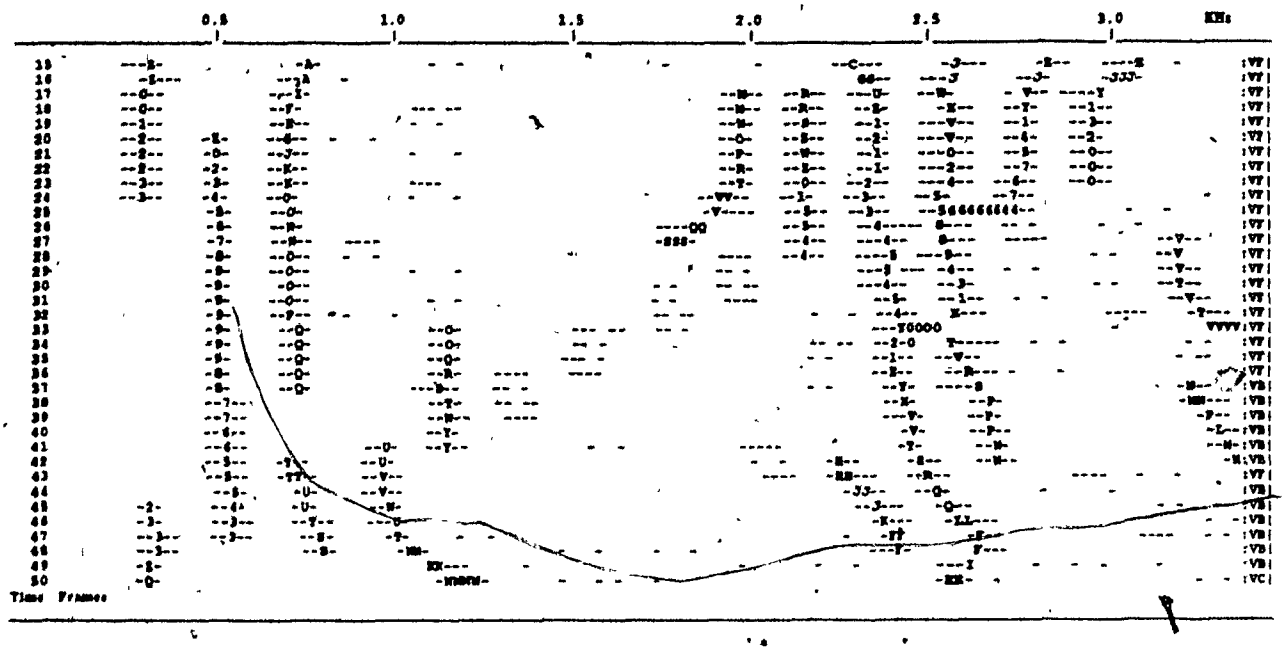


Fig. 4.8 Example of frame based labeling using CPMM. Labels (VB = back vowel; VC = central vowel; VF = front vowel) are attached at the end of each frame.

The a-priori probability returned by CPMM is translated into symbols of the alphabet § and the best candidate is kept in a vector of vocalic symbols, Vs, where Vs_{ij}, the j^{th} element of Vs contains the vowel identified for the internal segment, i, at the time frame, j. When the analysis is completed for the entire internal segment, the vector Vs is scanned to select the best vocalic candidate. The selection criteria is based on finding the candidate which is present for a maximum number of times sequentially in the vector. If no candidate appears to be present consecutively for a reasonable length of time, then the segment will be considered as non-sonorant and reports back to the supervisory network.

Fig. 4.8 shows an internal segment with vocalic symbols attached at the end of each time frame.

Even though frame analysis may cause certain overhead in processing, the following reasons will show why it is important to employ such a task:

1. If vocalic candidates identified by the CPMM are going to play a significant role in the final hypothesis stage, then robust recognition of these elements are absolutely necessary.
2. Frame analysis are performed only on internal segments and, usually, internal segments are rather short (≈ 20-30 time frames).

3. Unlike, confirm_vowel_rules, frame analysis is not applied on all segments. It is applied only in cases when the absolute difference of a-priori probability between two candidates is very small, $\delta < 1.0$.

The results obtained directly by CPMMs were above 90% as reported in [34]. With the application of rules and frame analysis technique, the final recognition rate for detecting place-of-articulation was close to 99%.

4.6 Chapter Summary

In this chapter we discussed in detail some of the applications of perceptually described elements in higher level processing. Some of the ideas presented were:

- Internal segmentation of sonorant regions from speech patterns.
- Internal segments are quasi-stationary which contain an acoustic event that occurred within a short-time period.
- Concatenation of acoustic events in several short-time periods would provide global information about the behavior of an acoustic segment (events like transition in place-of-articulation etc.).
- Perceptual components within internal segments can be used for learning and subsequent recognition of an acoustic event within a short-time period using statistical models like HMMs.
- A new statistical model called CPMM, which uses parameters in frequency domain, rather than the conventional time domain, has been proposed.
- The proposed CPMM was used to learn and identify place-of-articulation of sonorant regions.
- Possible errors generated by the CPMM were corrected using rules based on acoustic knowledge and by applying CPMMs on every time frame.
- Over 99% recognition was obtained on place-of-articulation detection for a large number of speakers over a variety of context by successfully implementing the proposed techniques.

Chapter 5

Identification of Vowels and Diphthongs using Perceptual Components

In this chapter we discuss in detail the recognition of vowels and diphthongs using perceptually organized components which have already been detected and described. Vowels and diphthongs are present in the sonorant regions of the signal which are either stationary or quasi-stationary. Only regions of this type were considered in the pattern analysis.

Even though extensive work has been done in the past on vowel recognition, very little has been done for diphthong recognition. Vowels appear as diphthongs in an acoustic segment if there are transitions in the place-of-articulation and (or) in the manner-of-articulation. Correctly identifying these transitions and distinguishing them from vowels would greatly enhance the performance of any recognition model. A classical example is the letter/digit recognition system. In this system, each letter/digit has a consonant-vowel combination or just pure vowels. In cases where there is consonant-vowel, (eg. "k"), the plosive sound of "k" is followed by the diphthong sound /ei/. The confusion of "k" among other plosive sounds can be solved if the diphthong in "k" can be recognized properly.

The task is to recognize vowels and diphthongs in any given sonorant region in any given context. Before we look into the recognition process, some detailed discussion about diphthongs and their properties are looked into.

5.1 Acoustic Properties of Vowels and Diphthongs

Vowels

Vowels are characterized by substantial energy in the low- and mid-frequency regions. The energy in the high frequencies, above 3.5 KHz, are usually not considered. Vowel recognition algorithms often rely on the values of the first three formants in order to differentiate vowels. For example front vowels are characterized by a large difference between first and second formant values.

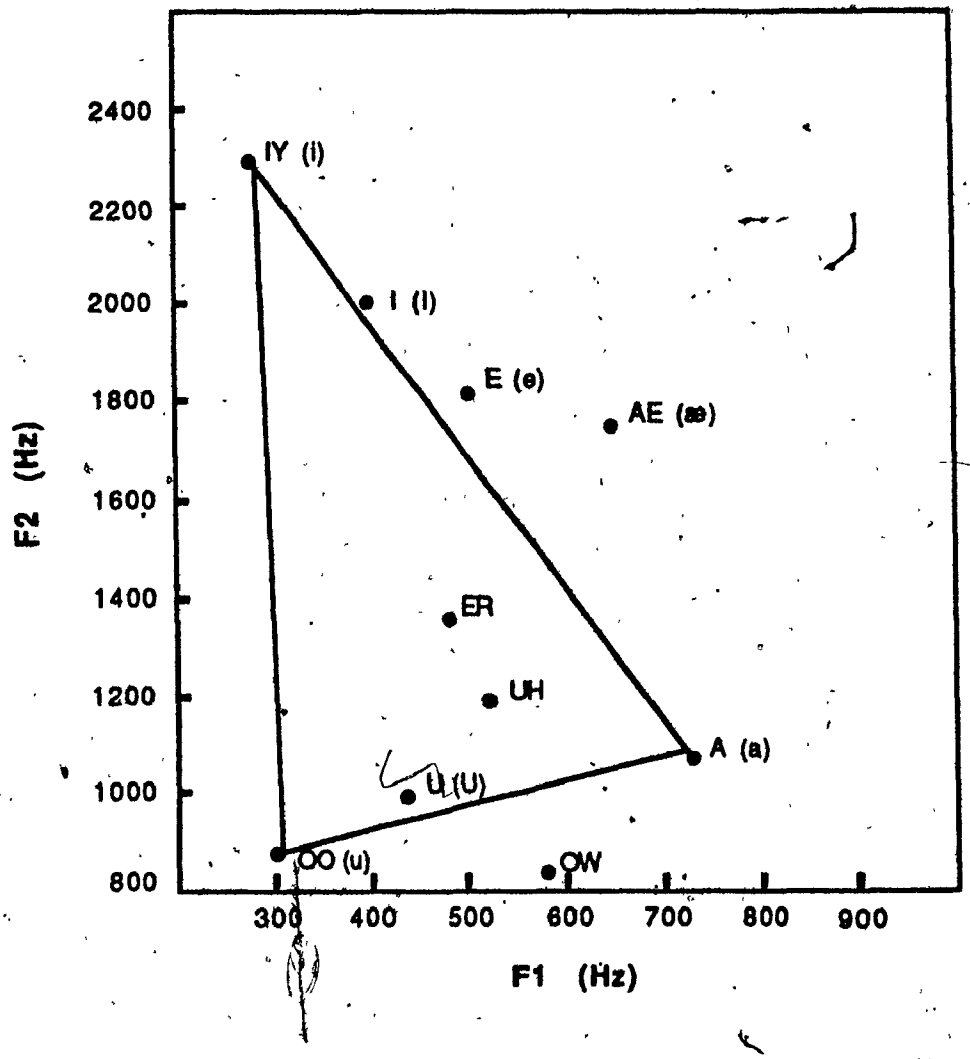


Fig. 5.1 The vowel triangle (After Rabiner and Schafer [8])

Spectral characteristics of vowels are heavily influenced by adjacent consonants especially if the consonants are glides or liquids. Fig. 5.1 shows the plot of the second formant frequency versus the first formant frequency for the vowels of the American English, and Table 5.1 shows approximate values for vowel formants F_1, F_2, and F_3 for a male speaker.

Diphthongs

There has been disagreement among Phoneticians about what is and what is not a diphthong since many vowels may appear as diphthongs in American English. A

reasonable definition of a diphthong is that "a diphthong is a gliding monosyllabic speech unit that starts at or near the articulatory position for one vowel and moves to or toward the position for another"[99]. The diphthongs in continuous speech often do not achieve the second steady state mostly due to the coarticulation caused by the following consonant.

Table 5.1 Typical Formant values for Vowels

FORMANT FREQUENCIES FOR THE VOWELS					
type written symbol for vowel	IPA symbol	typical word	F1	F2	F3
IY	i	beet	270	2290	3010
I	ɪ	bit	390	1990	2250
E	ɛ	bet	530	1840	2480
AE	æ	bat	660	1720	2410
UH	ʌ	but	520	1190	2390
A	ɑ	hat	730	1090	2440
OW	ɔ	bought	570	840	2410
U	ʊ	foot	440	1020	2240
OO	u	boot	300	870	2240
ER	ɜ	bird	490	1350	1690

A phonetic recognizer for diphthongs must be capable of considering rate of change of parameters explicitly, since steady state formant values may or may not be available or even reliable.

Holbrook and Fairbank [99] have done extensive study on diphthong formants and their movements. One of the key features of diphthongs is that the place-of-articulation changes at some point in time. These changes are clearly visible along formant contours. However, correct recognition of formant contours when sharp transitions occur, is a problem in itself.

In American English, there are six diphthongs[99], namely:

/eI/	⇒	bay	/oU/	⇒	boat
/aI/	⇒	buy	/aU/	⇒	how
/oI/	⇒	boy	/ju/	⇒	you

In the case isolated letters and digits, there exists two more diphthongs called,

/ua/	⇒	one	/eo/	⇒	zero

Fig. 5.2 shows the temporal variations in frequencies and energies of formants during the pronunciation of diphthongs. Diphthongs can be basically divided into two major groups, the /i/-diphthongs consisting of /ei/, /ai/, /oi/, and the /u/-diphthongs consisting, /ou/, /au/, and /ju/. Certain unique properties are present within each subgroup, for example:

/i/-diphthongs (/ei/ /ai/ /oi/)

- first formant, F1, falls from ≈ 500 Hz to ≈ 400Hz.
- second formant, F2, rises from ≈ 2000Hz to ≈ 2200Hz.
- energy of formants initially increases and decreases towards the end.

/u/-diphthongs (/ou/ /au/ /ju/)

- first formant falls
- second formant falls
- energy, from beginning to the end, decreases.

Both the rising and falling properties of formants will occur at the transition stage, that is, when the place-of-articulation changes. The transitions of diphthongs in F1 and F2 plane is shown in Fig. 5.3.

The properties discussed above are based on analytical concepts in ideal situations. In reality, the second and third formants are difficult to detect when transitions occurs. Also, in many situations, formants may not appear as smooth contours and in these situations, formant tracking algorithms may even fail. Fig. 5.4a and 5.4b show the patterns which contain diphthong /ai/. In the first case, the situation is ideal whereas in the second case, it is difficult to see the transitions.

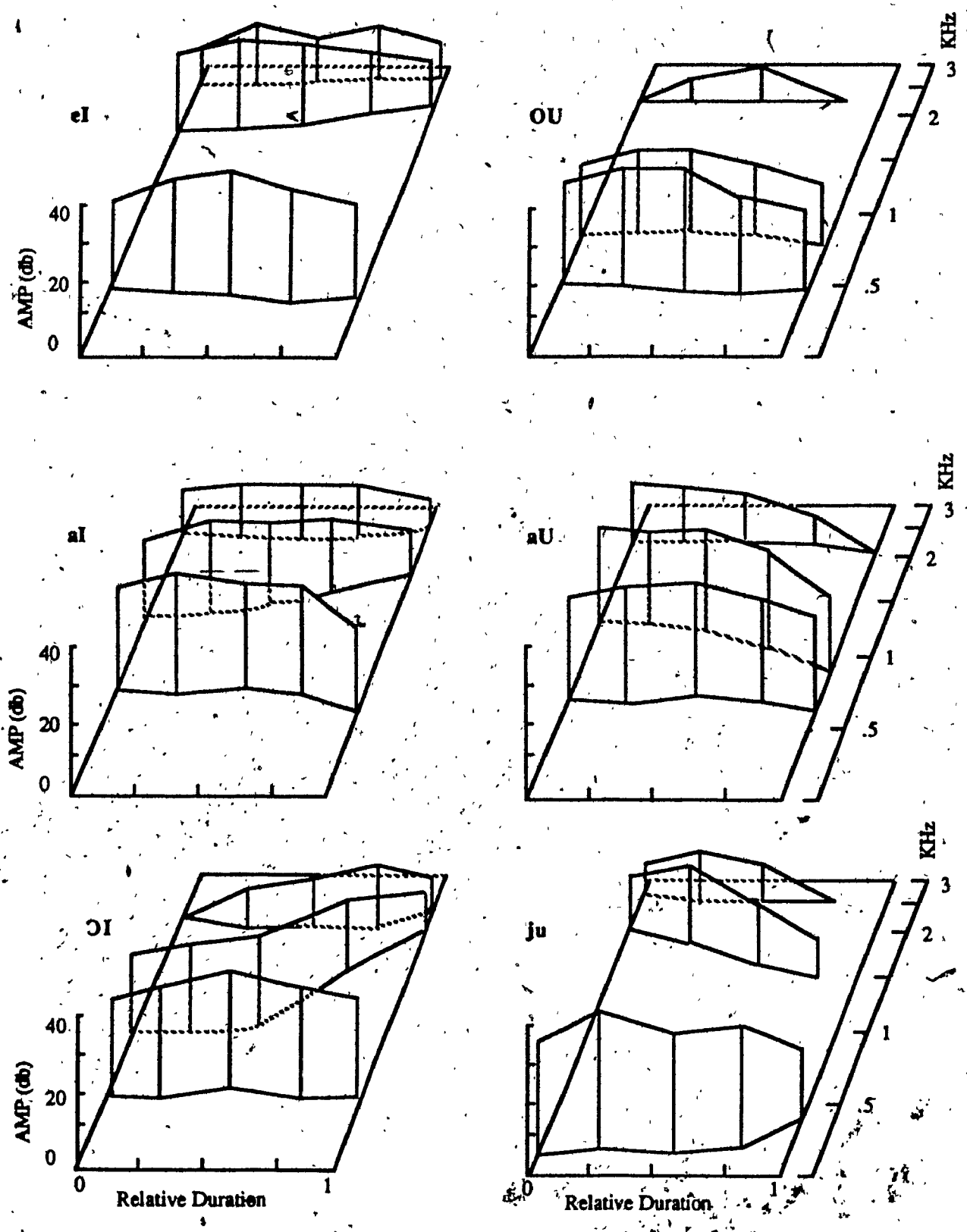


Fig 5.2 Temporal variations of median frequencies and amplitudes of formants during the course of utterance of diphthongs [99].

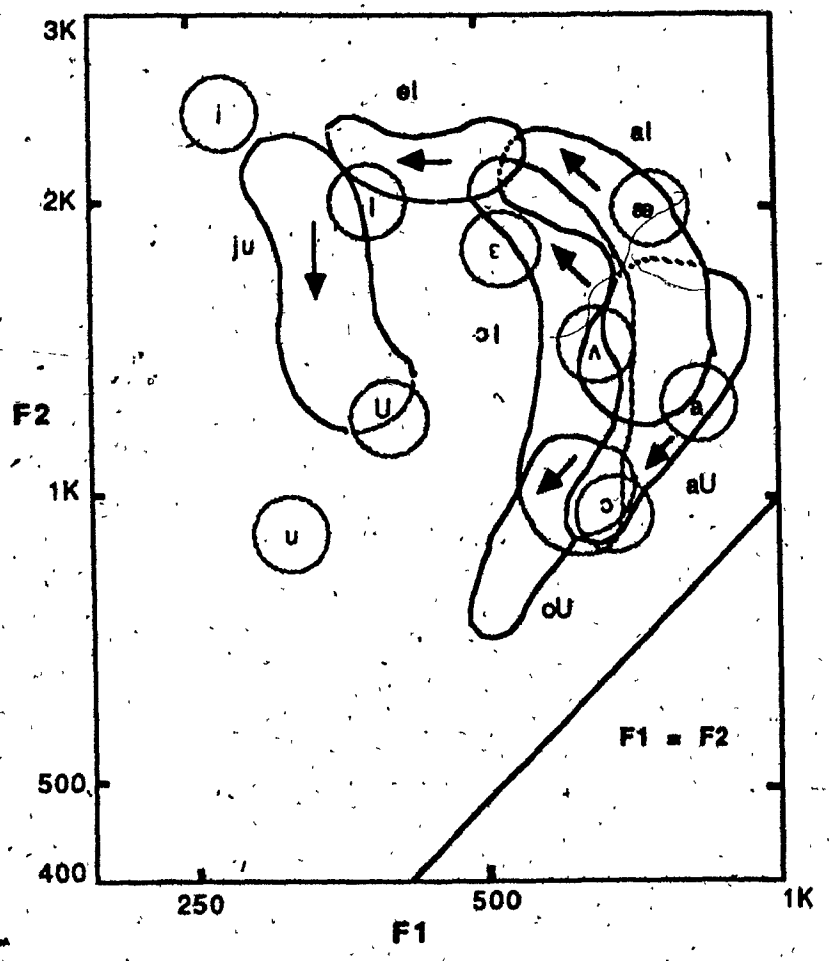


Fig. 5.3 Time variations of the first two formants for diphthongs [8].

5.1.1 Perceptual Properties of Vowels and Diphthongs

The properties which are detected and described as perceptual components for vowels and diphthongs are somewhat different from those properties which are detected and described based on analytical methods. Some of the most frequently observed perceptual properties for vowels and diphthongs are illustrated in Table 5.2. Each vowel or diphthong listed in Table 5.2 has one or many description symbols belonging to the set, $\{ \Gamma_1 \Gamma_2 \Gamma_3 \}$. A "*" followed by a property symbol indicates that there may be more than one symbol of the same type present. Note that in the sample patterns of Table 5.2, the frequency grows along the horizontal axis and the vertical axis represent time frames.

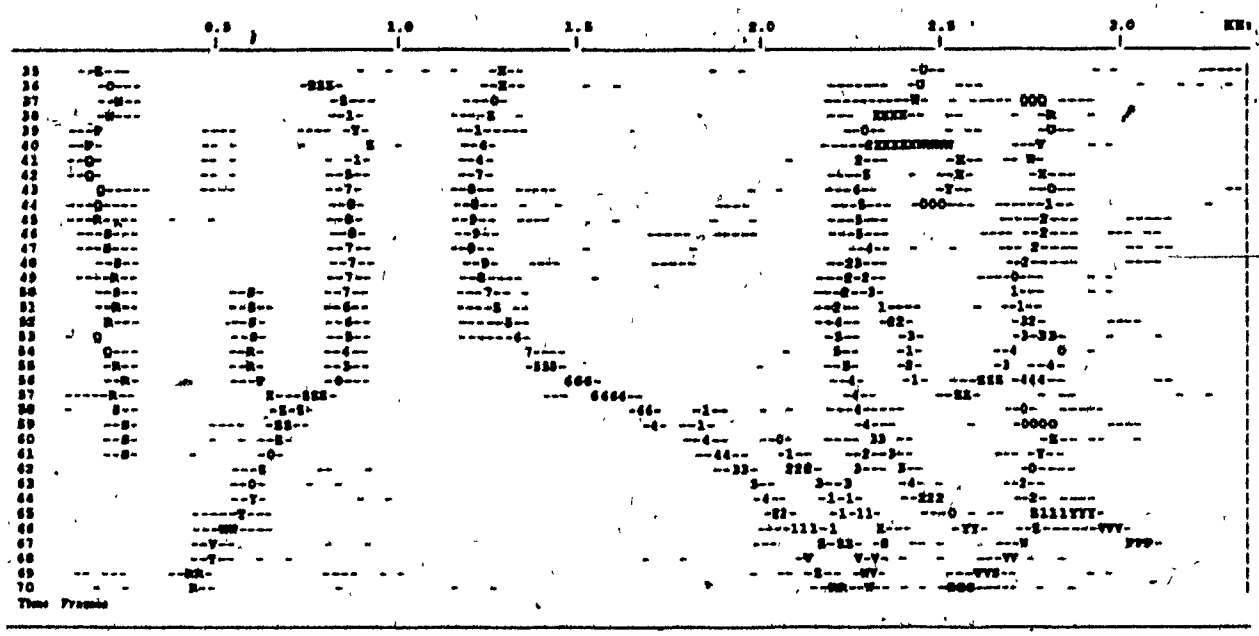


Fig. 5.4a An ideal pattern of a diphthong /ai/ in letter "i"

Property symbols in Table 5.2 are detected by algorithms introduced in Chapter 4 and that place-of-articulation is detected from spectral lines using CPMM. There are formant lines as well as some noisy lines among these spectral lines. Property symbol column contains regular expression of most likely sequence of detected Level-2 and Level-3 descriptors.

5.2 Recognition of Diphthong and Vowel like Phonemes Using Perceptual Components

In classical Knowledge Based approach, vowels are recognized based on rules applied on pure locational as well as quantitative information about the first 3 formants. Since these rules heavily depend on the numerical values, if formants are not detected properly, or if they are missed or mislocated, it is possible that a wrong hypothesis would be made.

An approach which does not depend on formants to determine place-of-articulation as well as transitions is most appropriate to handle such situations. Some of the perceptually organized components can be effectively used to solve problems of this nature.

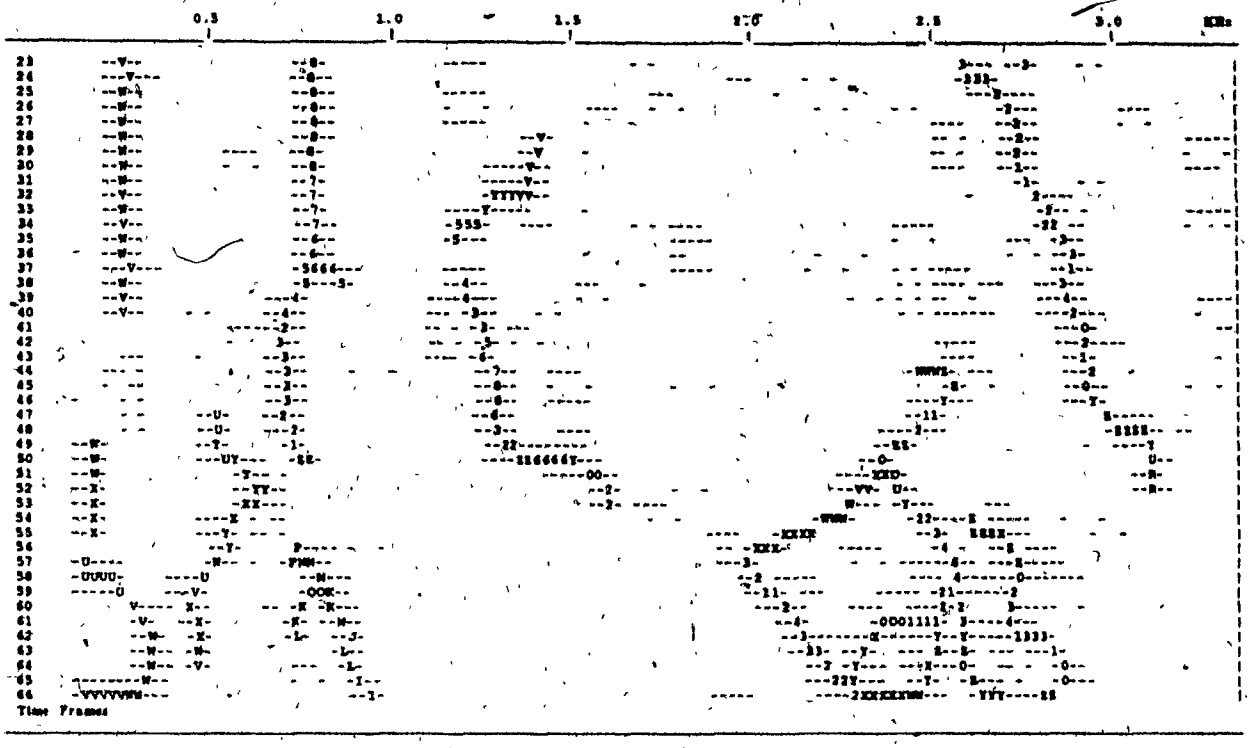


Fig. 5.4b A distorted pattern for a diphthong '/ai/ in letter "i"

In our approach, formants are not detected but location and qualitative information about spectral lines is learned using statistical methods which are capable of delivering the proper hypothesis even in the presence of distortions.

The classification and recognition procedures effectively use hierarchically organized perceptual components at various levels including the output from CPMM. A flow graph of the overall recognition strategy is shown in Fig. 5.5.

The vowel and diphthong recognition level is shown in detail in the flow graph of Fig. 5.6. Elaboration on each item of Fig.5.6 is considered now.

All description symbols generated at various levels in the hierarchy are kept in vector, Z, where, Z is of the form,

$$Z = (\zeta, t_b, t_e, f_{beg}, f_{end}, \varphi),$$

where,

ζ consists of descritpions $\in \{ \Gamma_1, \Gamma_2, \Gamma_3 \}$

(recall that, $\Gamma_1 = \{$ ASND, DSND, STCR, STCL, FDN $\}$,

$\Gamma_2 = \{$ NSPH, NSPT $\}$,

$\Gamma_3 = \{$ VB, VC, VF $\}$)

t_b is the beginning time of the property symbol in ζ

t_e is the ending time of the property symbol in ζ

f_{beg} is the beginning frequency of the property symbol in ζ

f_{end} is the ending frequency of the property symbol in ζ

$\wp$ is a measure of strength associated with each property at the time of extraction as described in section 3.5.4.

Table 5.3 shows a sample of vector Z, generated for an acoustic segment.

The vowel, diphthong recognition process accepts vector, Z, as the input and generates pre-conditions, performs classification based on pre-conditions, applies detailed-rules based on pre-conditions, extracts detailed properties based on already present knowledge about the speech segment, and finally, makes the hypothesis.

5.2.1 Pre-condition and Classification

The first step towards hypothesis generation is establishing a set of pre-conditions. The pre-conditions are formulated by considering the Γ_3 set in ζ, in property vector, Z. Since we allowed over segmentation during the internal segmentation process, and type Γ_3 symbols are generated for each internal segment, it is likely that some of the symbols generated are the same for successive internal segments.

Pre-conditions are, all possible sequences which can be generated by the CPMM for any given speech segment. In order to reduce the search space, the vocalic sequence generated by the CPMM is encoded in the following way:

Symbols of the same type that appear suecessively can be collapsed into one single symbol. For example, if the sequence generated is,

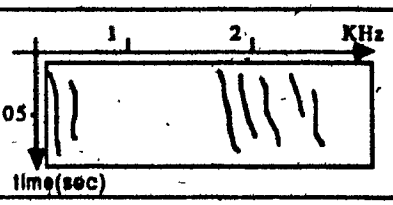
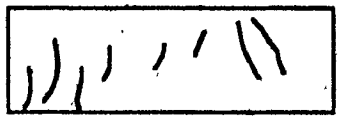
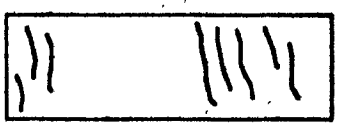
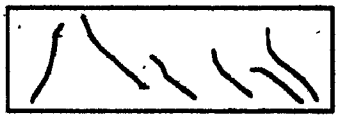
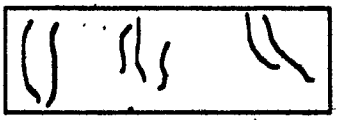
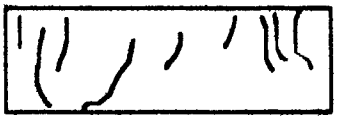
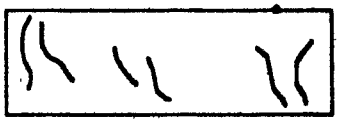
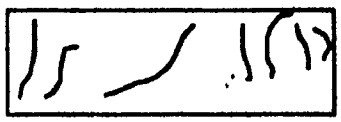
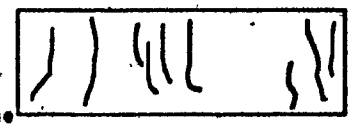
$$\langle VF\ VF\ VB\ VC\ VC \rangle$$

then the sequence can be reduced as,

$$\langle VF\ VB\ VC \rangle$$

This encoding process reduces the total number of possible sequences to a very small one for all the vowels and diphthongs under consideration. The encoded sequences are used as pre-conditions for next-stage processing. Based on the detected sequence, each possible sequence is considered as a class and assigned a unique class number. Table 5.4 shows various allowable sequences and their class designation. This is strictly an implementation detail.

Table 5.2 Perceptual properties of Vowels and Diphthongs

sound	property symbol	comments	stylized patterns
/I,i/	VF*	mid-band regions contain no lines	
/o,u/	VB*	/o/ may appear as /oU/ and /u/ as /eU/	
/ei/	VF*, FDN	there will be more than one VF detected	
/ai/	VC*VF* ASND or STCR DSND or STCL	ASND originates from mid-band and from begining frames	
/au/	VC*VB*	/o/ pronounced as dipthong	
/ju/	VF*VB* DSND or STCL	strong DSND or STCL originates from mid-band and from middle of the segment	
/ua/	VB*VC*	unique case for "one"	
/eo/	VF*VB* DSND	unique case for "zero" when "r" is missing	
/a/	VC*	several strong lines in mid-band region	

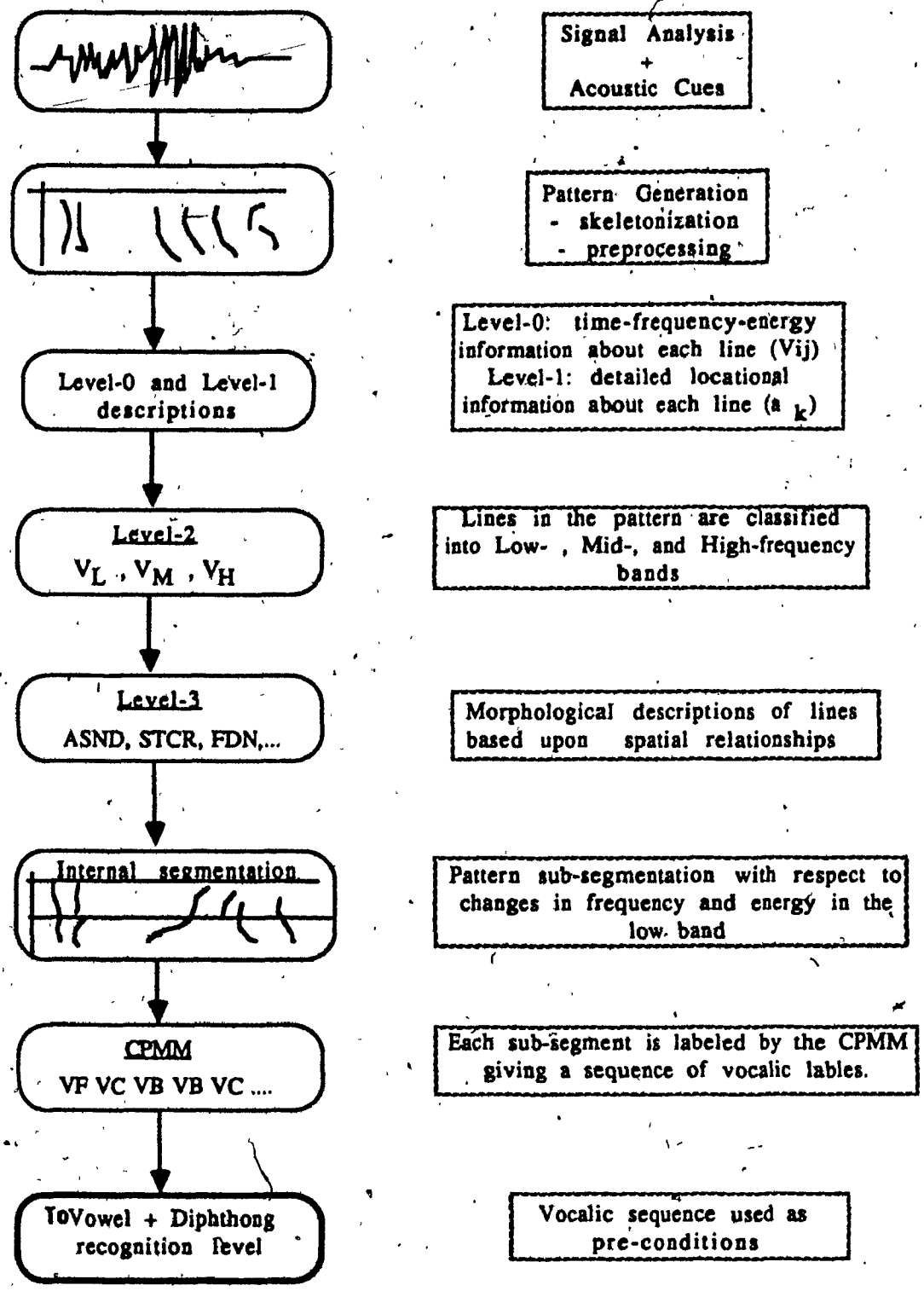


Fig 5.5 Flow graph of the Speech Pattern Analyzer (SPA)

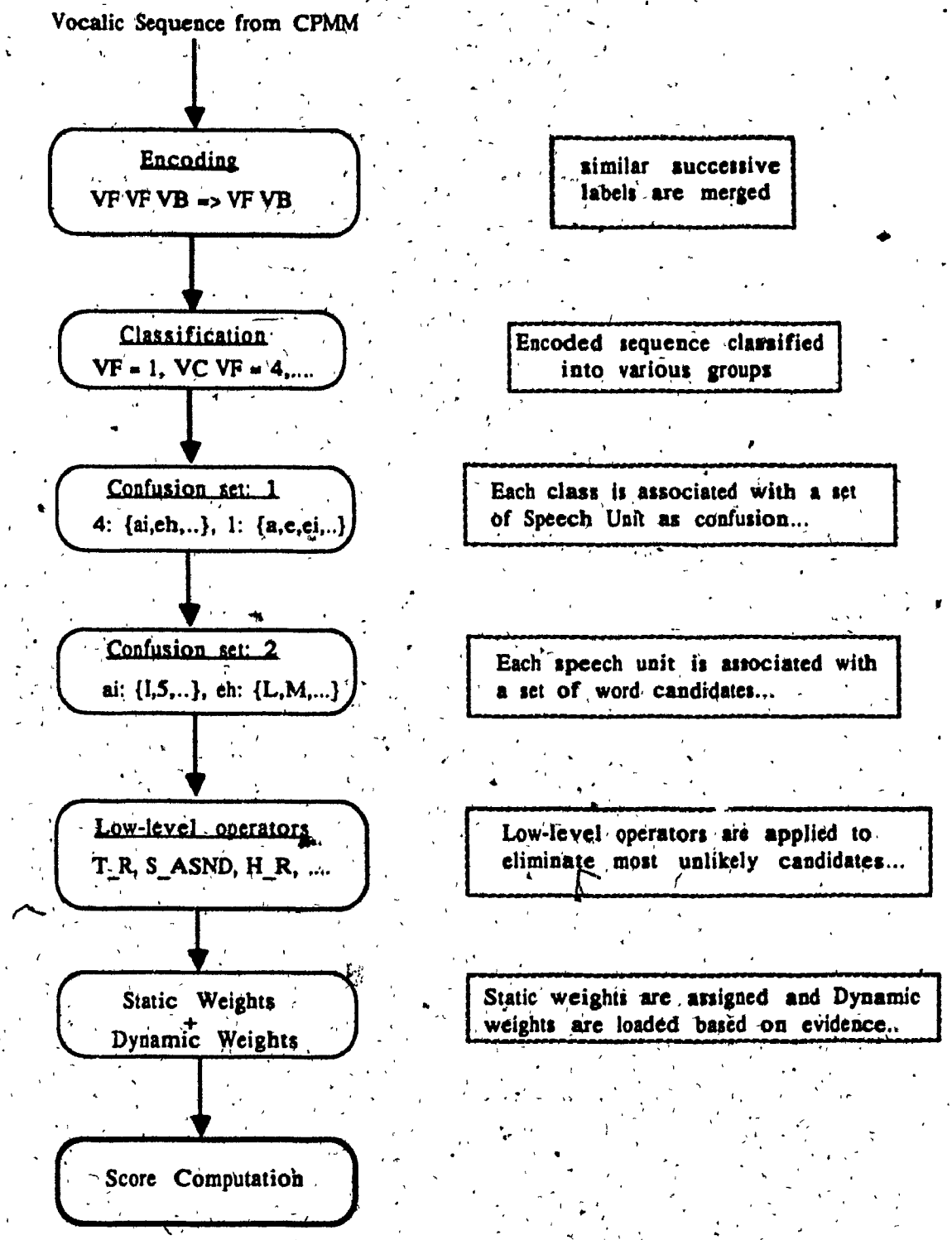


Fig 5.6 Flow graph of Vowel and Diphthong identification scheme.

Table 5.3 A sample output of the vector Z generated by SPA for the letter "y"

```
 FDN ( 38,  50,1.000)
ASND (  0,   0,   0)
DSND (  0,   0,   0)
STCL ( 27,  33, 891,  891,0.6947)
STCL ( 38,  50, 891,  891,0.6947)
STCL ( 48,  66, 702,  729,0.5579)
STCR ( 33,  48,1512,1512,0.8482)
STCR ( 47,  53,1728,1728,0.7422)
STCR ( 50,  66,1917,2214,0.6209)
STCR ( 54,  64,2160,2349,0.5689)
 V_B ( 25,  35,  756,  702,8.8197,5.9429,6.6806, 0.3425)
 V_C ( 38,  49,  891,  891,8.2992,8.3738,8.4878,13.7220)
 V_C ( 50,  55,  702,  702,8.4283,8.6412,8.4804, 2.5494)
 V_F ( 56,  61,  486,  513,7.8895,0.0000,8.4188, 1.6192)
```

Table 5.4 Symbol sequences and class designation

Number of Symbols	Symbol Sequence	Class Number
1	VF	1
1	VC	2
1	VB	3
2	VC VF	4
2	VF VB	5
2	VB VC	6
2	VF VC	51
2	VB VF	52
3	VB VC VF	8
3	VF VC VF	53
3	VF VC VB	54
3	VF VB VF	55
3	VC VF VB	56
3	VB VF VC	57
3	VB VC VB	58
3	VC VF VC	59
3	VC VB VC	59
4	VF VC VB VF	9
4	VF VB VC VB	61

5.2.2 Allocation of Confusion Set

To each class shown in Table 5.4, a possible confusion set of "speech units" is attached. The confusion set consists of all sound class in vowels and diphthongs represented in an orthographical way, using the symbol, $\Im$, and defined as,

$$\Im = (a,e,ai,o,eu,uai,ua,eh,ah,eo,nai,au)$$

Each symbol in the set belongs to either a vowel, diphthong, or a vowel-diphthong-like sound. Note that these symbols are not necessarily the same as the IPA representation or the ARPAbet representation of phonemes. Complete alphabets in English and the numerals can be associated with one of the above symbols. The confusion set allocation is strictly based on the vocalic-sound-class it belongs to.

There is also an association between the symbols in $\Im$ and the symbols in Γ_3 generated by CPMM. This association together with their class number and the confusion set is shown in Table 5.5.

Table 5.5 Class, Pre-condition sequence, and Confusion sets

Class Number	Pre-condition Sequences	Confusion set (Speech Unit)
1	VF	e
2	VC	ah
3	VB	o
4	VC VF	ai
5	VF VB	eu
6	VB VC	ua
7	VC VB	au
8	VB VC VF	uai, ua
9	VF VC VB VF	nai
0	VF	ei, a
51	VF VC	eh
52	VB VF	ua, uai
53	VF VC VF	ai, uai, eh
54	VF VC VB	eh, au
55	VF VB VF	eu, ua
56	VC VF VB	eu, au, ua
57	VB VF VC	uai, ua
58	VB VC VB	uai, ua, ai, nai
59	VC VF/VB VC	ah, eh

Each item in $\mathfrak{S}$ can be associated with a confusion set of words in the vocabulary, in this case, all the letters and digits, as shown in Table 5.6.

Table 5.6 Letter/Digit confusion sets of Speech Units

Speech Units	Letter/Digit Confusion set
æ	A, 8, J, K
æi	A, 8, J, H, K
i	B, C, D, E, G, P, T, V, 3, 6, Z
ai	I, Y, 5, 9
eh	F, H, L, M, N, S, X, 7
o	O, 0, 4
u	U, 2
eu	Q, U, 2, W
ɹh	R
ua	1, Y
nai	9, Y, I
uaɹ	Y, 1, I

The above classification is an ideal one in the sense that, in reality, being in the right class will depend on the pre-condition sequence generated by CPMM. Because of the speaker variabilities, two things may happen to the anchor line, £, which in turn will affect the output from CPMM:

1. In the case of diphthongs, if the head part or the tail part is pronounced weakly, then there may not be any significant shift in either frequency or energy in £. This will cause no internal segmentation possible. An example of such a situation is diphthong /æi/. The head part of this diphthong is /æ/ and the tail is /i/. If the sound is spoken properly, an observable transition would be present in £ when the sound changes from /æ/ to /i/. However, if either the head or tail is pronounced weakly, the CPMM would generate either pure VFs or pure VCs.

2. In the case of single vowels, a sound pronounced with variations would cause changes in £ if there are variations in frequency and energy. These variations would result in internal segmentation and each internal segment would cause the CPMM to generate a vocalic symbol. Again, if the variations are small, the symbols generated by CPMM would be of the same type and would be collapsed into one by the

encoding process. However, if the symbols differ, this would signal a different set of pre-conditions which would result in classifying the sequence into a different class, possibly even a diphthong class.

If any of the above mentioned problems occur, this would result in classifying a single vowel as diphthong or a diphthong as a single vowel. Both of the above situations are heavily speaker dependent. From the observation of a large amount of patterns, the situation where a diphthong is classified as a single vowel occurs more frequently than a single vowel being classified as a diphthong. Solving problem 1 is easier than solving problem 2, since, in problem 1, one can look for extra features of diphthongs or use already detected features like ASND or DSND to reconfirm the output of CPMM.

5.2.3 Head-Tail Verification and Class Adjustment

As a first level approach to correct any loss of information which could have been caused by internal segmentation, a head-tail analysis is done on a pre-selected set of classes. The selection was based on those classes which should really belong to another class, if no loss of information would have occurred.

For example, if the class obtained corresponds to single vocalic of type VF, and if there exists another vowel symbol, VB, at the tail part of the segment which was missed, then detection of this tail would change the class to VF VB corresponding to the diphthong /eu/. Changes of this nature, possible under certain allowable sequences are listed in Table 5.7.

Table 5.7 Possible distortion sequence for Vocalic symbols ("*" means symbol may be repeated)

Distorted Sequence	Sequence after Head/Middle/Tail-recovery
VF*	VF* VB
VF*	VC VF*
VC*	VB VC*
VC	VC VB VF
VC	VC VB
VC	VC VF
VB	VF VB
VC VF	VB VC VF
VC VF	VF VC VF
VB VC	VB VC VF

The detection of head-tail information is based on frame-analysis technique explained in section 4.5.2. The Head-Tail-Analysis Algorithm, given below, shows how detailed testing on frames could retrieve missed information. The algorithm is performed only on those classes which are listed in Table 5.7.

```
Algorithm Head_Tail_Analysis (HTA_Algm)
begin
    if class in [class-subset] then
        begin
            if head to be analyzed then
                    head := frame_analysis(t_b, t_b + α, S_m)
                    if head then adjust_class
            else
                    tail := frame_analysis(t_e - α, t_e, S_m)
                    if tail then adjust_class
        end
end (HTA_Algm)
```

Comments on HTA_Algm

- t_b, t_e are the time beginning and time ending of the acoustic segment.
- α is an offset value which determines how many frames have to be considered for frame analysis.
- for both head and tail analysis, α, is set at value 10.
- S_m is a symbol belonging to § and tells frame analysis function the type of head or tail (VF or VB or VC) requested.
- function frame_analysis returns true if the type of head or tail requested is found for at least more than half of the total number of frames analyzed.
- If head or tail is true, then the class is re-adjusted to the new class.

The head-tail processing strategy performed extremely well in the cases where the sound produced was a diphthong and the internal segmentation was unable to catch the transition. The diphthong /eu/ is a typical case in which the anchor line evolves smoothly down without showing any changes with respect to frequency or energy.

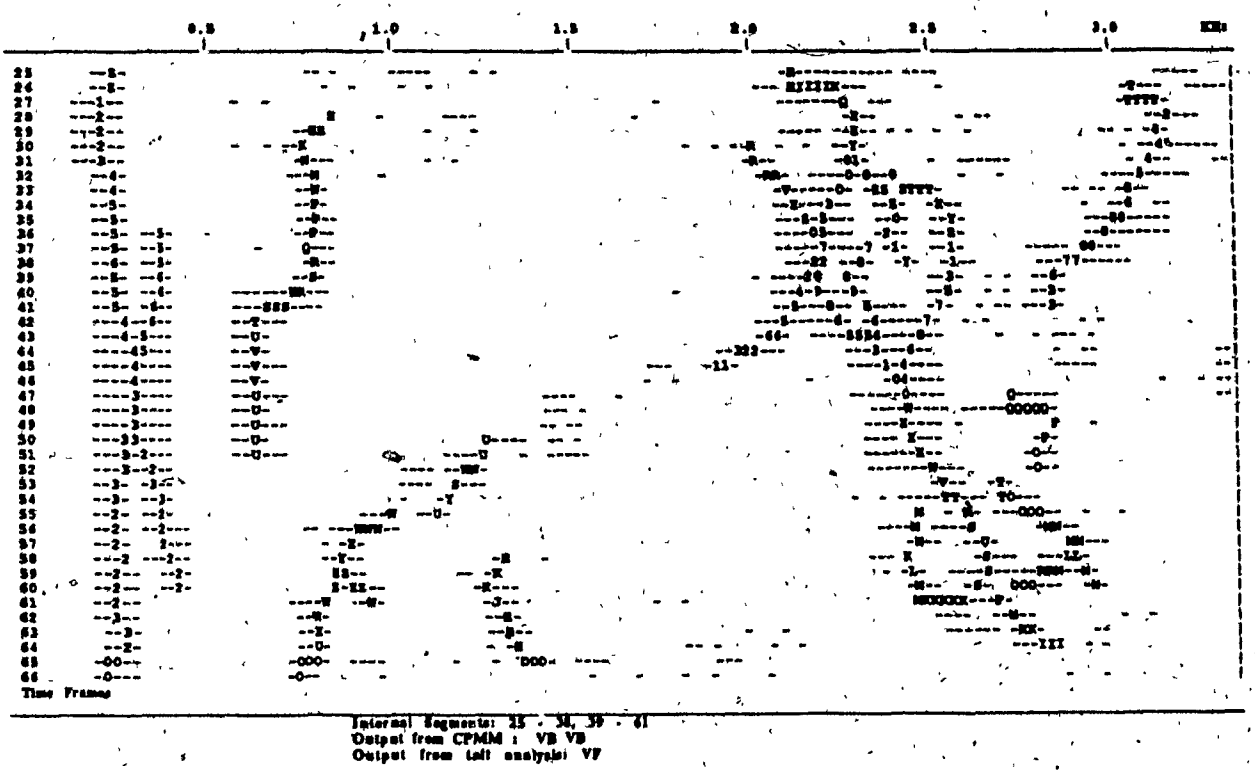


Fig. 5.7 An example of "head" recovery using frame analysis for letter "u"

Usually, the string produced in this situation would be a sequence of VBs belonging to class 2. The head-tail analysis recovers the VF head and changes the class to 5. Fig 5.7 shows the pattern of a /eu/ with output from CPMM, the initially assigned class, the result of head-tail analysis, and the resulting class.

5.3 Diphthong and Vowel Hypothesis Generation

In order to make the final hypothesis, measures are computed for each candidate in the set, $\Im$, using a dynamically adjustable weight array, T. Array T is made of elements:

$$T_{ij} = W_{ij}$$

where,

$i \in \Im$,

$j \in \zeta$, and

W_{ij} is the weight associated for each i, j entry in the array, T.

Initially, array T is set with static weights for each i,j entry. These static weights are based on expected properties for each candidate. In reality, the detected properties will differ from the expected properties; more than one candidate may

satisfy the detected properties. The initial static weights would provide a fair score to all the candiates with the expected properties.

The static weights are derived from prior knowledge about the acoustic and phonetic properties of each candidate. In the later stage, the weights are dynamically adjusted to give maximum weight to the most probable candidate. The dynamic weight adjustment is performed using the pre-conditions and certain low level operators applied to the acoustic segment. Table 5.8 shows I_r with its initial static weights.

Table 5.8 The Initial Static Weight Table.

Properties								Speech Unit	Vowel Symbol sequence
VF	VB	VC	FDN	ASND	DSND	STCL	STCR		
0.4	0.0	0.0	0.3	0.3	0.0	0.0	0.3	a	VF
0.4	0.0	0.0	0.3	0.3	0.0	0.0	0.3	e	VF
0.4	0.0	0.0	0.3	0.3	0.0	0.0	0.3	ei	VF
0.0	0.4	0.0	0.3	0.3	0.3	0.3	0.3	o	VB
0.0	0.4	0.0	0.3	0.3	0.3	0.3	0.3	u	VB
0.4	0.0	0.4	0.3	0.3	0.3	0.3	0.3	ai	VC VF
0.4	0.4	0.0	0.3	0.3	0.3	0.3	0.3	eu	VF VB
0.4	0.4	0.4	0.3	0.3	0.3	0.3	0.3	ua	VB VC VF
0.0	0.0	0.4	0.3	0.3	0.3	0.3	0.3	ah	VC
0.0	0.0	0.4	0.3	0.3	0.3	0.3	0.3	eh	VC
0.4	0.4	0.4	0.3	0.3	0.3	0.3	0.3	uai	VB VC VF
0.4	0.4	0.4	0.3	0.3	0.3	0.3	0.3	nai	VF VB VC VF

5.3.1 Dynamic Weight Adjustment (DWA)

The dynamic weight adjustment technique is used before computing the actual score for each candidate. The reasons for using dynamically adjustable weight table for score computations are,

- to provide the maximum score for the best candidates; i.e. the candidates which are close in their expected properties and detected properties, while maintaining a reasonable score for all the others by considering their detected properties, must receive the maximum scores. Such a score distribution is used by the supervisor procedural

network, which will be discussed in detail in Chapter 6, so that the final recognition process can consider other less probable candidates too.

- the vowel-diphthong hypothesis routine scores the candidates by multiplying the weight of each property against its detected measures (as it will be discussed in detail later) and adds them together. Since this operation is performed sequentially, there exists a synchronization problem with certain properties. For example, if the class detected is 5 (/ua/) then the vocalic sequence would be, ⟨VBVC⟩. During score computation, the weight and measure of VB and VC are considered. However, class 7 (/au/) with ⟨VCVB⟩, also contains the same vocalic symbols. The only difference is that such symbols are detected at different time. The static weights in both cases would be the same, but the unique class number has the inherent property of providing the event synchronization. The DWA algorithm pulls up the weights of the right class. In the above example, if the class was 5, then the weights for VB and VC for /ua/ would go up while weights for /au/ would remain the same.
- after applying DWA algorithm, the confusion within a class is reduced by lifting up the weight of the most probable candidate. In most cases, a single candidate is given the maximum weight while the weights of the other candidates are increased only with less significance.
- since the weights are represented in a tabular form, the numerical values can be changed easily without any need for recompilation of the software. Also new properties and new candidates in the confusion set can be easily inserted.

Weights are adjusted after applying a set of operators to a certain class of numbers. Based on the result of the operator, the weight may be increased or decreased for all or some of the candidates in the confusion set. If no operator needs to be applied, then, by default, the candidate in the class will receive a weight lift.

The operators perform certain detailed tests on already extracted properties. The operators which are in use currently with their functions are given below:

5.3.2 Low-Level Operators

1. Operator: T_R

Computes the energy-ratio at the tail part of the pattern in the high frequency range. This operator is applied only if there is a vocalic transition detected by the CPMM, i.e., if more than one different symbols generated. For example, diphthong /al/, will have symbols VC VF generated by CPMM. In this case the energy ratio is calculated at high frequencies, when the transition occurs from VC to VF. The purpose of this operator is to detect whether there is any drop in energy in the tail region of the pattern. The T_R operator is used for distinguishing between different candidates for which identical sequences for place-of-articulation hypotheses are generated. For example, for /r/, the vocalic sequence generated is similar to that of /al/, i.e. < VC VF >. However, in the case of /r/, there will be a sharp drop in the energy where <VF> is detected.

Table 5.9 Class numbers and Low_level operators applied

Class Number	Operators Applied
4	T_R, S_ASND
5	S_NSPH
6	S_ASND, T_R
8	S_ASND, T_R, H_R
53	S_ASND, H_R, T_R
54	S_ASND, H_R, T_R
55	S_ASND, H_R, T_R
56	S_ASND
57	S_ASND
58	S_ASND, H_R, T_R
59	F_R

2. Operator: H_R

This operator is similar to T_R, except that H_R finds the energy ratio in the head region. The context of application is also similar to that of T_R. One major difference between T_R and H_R is that, T_R analyzes energy drops while H_R analyzes rises in energy. The situations in which all the operators are applied is summarized in Table 5.9.

3. Operators: S_NSPH, S_NSPT

The properties, NSPH (Non-Spectral Head) and NSPT (Non-Spectral Tail) are already detected during the internal segmentation process. The operators, S_NSPH and S_NSPT, verify the strength of the non-spectral head or tail. The properties NSPT and NSPH are usually detected for most nasal and fricative sounds. In the case of fricatives, these head and tail properties are more significant than for nasals. Therefore, the operator in this context helps to enhance the weight associated with the vocalic part in case of confusions.

4. Operator: S_ASND

The operator S_ASND, is similar to the previous case in that it verifies certain details of the property, ASND, which has been detected earlier. The operator looks for the strength, originating and terminating positions with respect to time and frequency, of the property, ASND. This is an important operator to isolate diphthong /uai/ from other similar candidates. Even though ASND property is detected for several other candidates, the locational information as well as the degree of slanting is unique for each particular diphthong.

The weight adjustments are done using a coding technique at the same time as the operators are being applied. The operators are not applied to all the classes, but new operators can be invented or more operators can be applied to any existing class, if needed, with great flexibility.

The weight adjustment algorithm, examines the class number, decides whether to apply operators or not, and assigns a code number or a set of code numbers to each class. Each code number is associated with a set of properties and weights which are maintained on a table. The weights appearing in this table can be adjusted externally.

These weights are loaded into T by the scoring algorithm and are used to compute the final score. Therefore, individual weights can be pulled up or down based on the outcome of the operators verification. The algorithm that applies operators and assigns a code set is given below:

5.3.3 Algorithm: Dynamic Weight Adjustment (DWA_Algm)

/ input to DWA_Algm is the detected Class number, C_I)/

1. test class number, C_I.
2. if operators need to be applied,
   then apply operators, from [T_R, T_H, S_NSPT, S_NSPH, S_ASND]
3. assign set of code numbers.
4. exit.

An example would clarify the algorithm in a better way. Given below is a pascal like code for class 8. Initially, class 8 consists of confusion set containing words "y" and "9".

```
Code_cnt = 1;
Case C_I of 8:
        begin
            if S_ASND then
                    Code_vect[code_cnt] = 801;
                    Code_cnt = Code_cnt + 1;
            if T_R
                    then Code_vect[code_cnt] = 802
                    else Code_vect[code_cnt] = 803;
                    Code_cnt = Code_cnt +1;
            if H_R
                    then Code_vect[code_cnt] = 804
                    else Code_vect[code_cnt] = 805;
    end;  {DWA_Algm}
```

Discussion on DWA_Algm

- Code_cnt is a counter which tells how many codes are assigned to the class C_I.
- Code_vect is a vector which stores all the code numbers, which are needed later for loading adjusted weights from code table, Tc.

numbers, 801, 802, etc are the actual code numbers. Each code number in code table, T_c has the format,

$$\{ C_l, C_{li}, S_m, W \}$$

where,

C_l is the class number,

C_{li} is the i^{th} code number in class C_l

S_m is a property symbol belongs to ζ, and

W is the adjusted weight.

for example, code 801 would like,

8 801 ASND 0.9

The full details of code table are given in Appendix-A.

5.4 Score Evaluation and Hypothesis Generation

Before computing the scores for each candidate, the weight array, T is updated with new weights, after applying the DWA_Algm. The weights associated with each code listed in Code_vec are read from the external code table, Tc, into weight table, T. All the other entries in T remain the same with static values assigned initially.

The final score for each candidate is computed using the formula,

$$S_i = \sum_{j=x}^{k} T_{ij} \times Z_j$$

where,

S_i is the score for the i^{th} candidate in score vector, S,

T_{ij} is the weight of i^{th} candidate and j^{th} property symbol,

Z is the feature vector with detected properties and their scores,

$i \in \Im$, is a symbol in the confusion set of speech unit,

$j \in \zeta$, represents a property symbol, and

$Z_j = \wp$ is the measure associated with each j.

For each candidate in set $\Im$, a score is computed and kept in vector S. The computed scores are not purely probabilistic since the measures associated with some of the property symbols are not based on any statistical evaluations. The score reflects a

measure of certainty based on several pieces of evidence.

In order to make the scores in S compatible with other scores elsewhere in the Procedural Network System (which will be discussed in the next chapter) the scores are normalized to 1.0 before returning the score vector. The normalization of the scores can be done as,

$$S_i = \frac{S_i}{\sum_{j=1}^{N} S_j}$$

where,

S_i is the i^{th} candidate in the score vector S,

N is the total number of candidates in the score vector.

After normalization, the score of each candidate will be between 0.0 and 1.0 inclusive. The score vector is returned to the Procedural Network in order to proceed with the word hypothesization process. Table 5.10 shows a typical score vector generated for the letter, "y".

Table 5.10 Score vector generated for letter "y"

UAI	=	0.1171
UA	=	0.1123
NAI	=	0.1003
AI	=	0.0915
EH	=	0.0867
AH	=	0.0867
EI	=	0.0525

5.4.1 An Illustration of the application of DWA_Algm

In this illustration we shall consider a real problem situation. The word spoken is letter "i" by a female speaker. After various levels of property extraction, the feature vector, Z, contains the following information for this particular utterance of the letter "i".

(STCL, 42, 63, 648, 621, 0.4896)

(STCR, 39, 44, 1701, 1701, 0.7619)
(STCR, 45, 64, 2106, 2376, 0.5783)
Z = (STCR, 46, 61, 2295, 2700, 0.5189)
(VC, 30, 42, 891, 891, 13.3369)
(VC, 43, 49, 675, 702, 1.7480)
(VF, 50,61, 486, 540, 2.8070)

When the feature vector Z enters the scoring phase, the following sequence of events will takes place:

1. Using the Γ_3 components in ζ of Z, pre-condition and class assignments are determined. In this example, the Γ_3 components are <VC VC VF>. Using the encoding process, the above sequence is reduced to <VC VF>. Using this sequence as the pre-condition, the class is assigned as 4, using Table 5.5 and Class 4 has the following confusion set:

{ai, uai, eh, ah, nai, ua} (a)

and each component in the above set has the following word candidates as confusion:

ai: {i, y, 5}
eh: {l, m, n, s, x, h}
ah: {r, i}
nai: {9}
uai: {y, i} (b)

The static weights are obtained from Table 5.8 for each item in (a) based on detected properties shown in Z.
Since there is no <VB> in Z, the static weights for {ai,uai,nai,ua} will be the same (properties: STCL,STCR,VC,VF and weights: 0.3, 0.3, 0.4, 0.4 respectively) and for {eh,ah} (properties: STCL,STCR,VC and weights: 0.3, 0.3, 0.4 respectively), the weights will be the same. At this point, all the word candidates with some expected properties are selected for the next stage analysis.

2. The next step is to apply the DWA algorithm. The important part here is the application of low-level operators, T_R, H_R, S_NSPH, S_NSPT, and S_ASND as explained in section 5.3.1. The low-level operators are applied

selectively for each particular class number (i.e. based on pre-conditions). In this example, the class is 4 and the operators applied are: S_ASND, T_R, and H_R. The results obtained after applying the operators are,

S_ASND = FALSE; T_H = FALSE; H_R = FALSE; (c)

Since S_ASND is false, the static weights for *ai* and *uai* are adjusted such that the weight of *uai* is decreased and the weight for *ai* is increased. The reason is that, S_ASND is a unique property for *uai* and it is not present in this case. H_R false means, *na* is unlikely and, hence, the weight for *na* is reduced. T_R false implies that, candidates in *eh, ah,* and *ua* are unlikely, and therefore, their weights are reduced. In both of the above cases, the weight of *ai* is increased. The weight adjustment is not made inside the software but they are already pre-adjusted and assigned with a code so that only the code needs to be entered while adjustments are made when the operators are applied during hypothesization.

3. During score evaluation, as explained in DWA_Algm, the weights associated with the selected codes are loaded from the weight table. In this example, the code 401 and the weights for each property in Z, having the code 401, are loaded. Refer Appendix-A for the symbols and their weights for code 401.
4. After score evaluation and normalization, the following scores were obtained for each candidate for this particular example,

ai = 0.2713; uai = 0.2062; eh,ah = 0.1526; nai,ua = 0.1087; (d)

showing *ai* as having the highest score. Notice also in (d), the candidates close to *ai* are also having a score.

5.5 Chapter Summary

- Properties of vowels and diphthongs based on analytical approaches and their drawbacks were discussed.
- Equivalent perceptual properties for vowels and diphthongs were established based on hierarchically grouped perceptual components.
- Search space was reduced by collapsing similar properties into one and by the application of pre-condition methods.
- Dynamically adjustable weights based on detected properties allowed the right candidates to be within the top 3 positions.

- Static weight association with all the candidates allowed less probable candidates to be included in the scoring process thereby avoiding accidental elimination of certain candidates in the event certain properties were not detected.
- Pre-condition rules, Static weight table, and Dynamic weight tables were maintained as separate entities independent of the software allowing easy modifications whenever necessary.
- The reasoning philosophy presented in this chapter follows a paradigm of hypothesis consolidation based on heuristics. Such a paradigm is used sometimes in Expert Systems. The advantage of such an approach is flexibility; the disadvantage is lack of a robust underlying theory. However, good experimental results obtained with the heuristic method suggests that a scoring technique based on probabilistic theory is worth looking into.

Chapter 6

System Performance and Experimental Results

In this chapter we show how the Procedural Network System described in Chapter 2 carries out the recognition task. Several practical examples from the vocabulary set will be shown with intermediate results which are returned by various sub-networks. A performance evaluation of the system will be made at the end showing the achievements stated in the design objectives.

6.1 An Example of Application

Let us assume we want to characterize sequences of letters and digits according to the lexicon defined in Table 6.1 with a little pause between them. Let us also assume we want to represent knowledge that is speaker-independent.

The PN conceived for this purpose has several levels.

Table 6.1 The 36 word vocabulary

Zero	Six	C	I	O	U
One	Seven	D	J	P	V
Two	Eight	E	K	Q	W
Three	Nine	F	L	R	X
Four	A	G	M	S	Y
Five	B	H	N	T	Z

The highest level, is represented in Fig. 6.1a. It consists of a "push" arc to a sub-network LEX representing the lexicon followed by an iterative jump to state *S0* under the condition "not-end" representing the fact that there is still a part of the input signal to be analyzed; otherwise the recognition process will stop on the execution of a POBABS arc with a stop function associated to it.

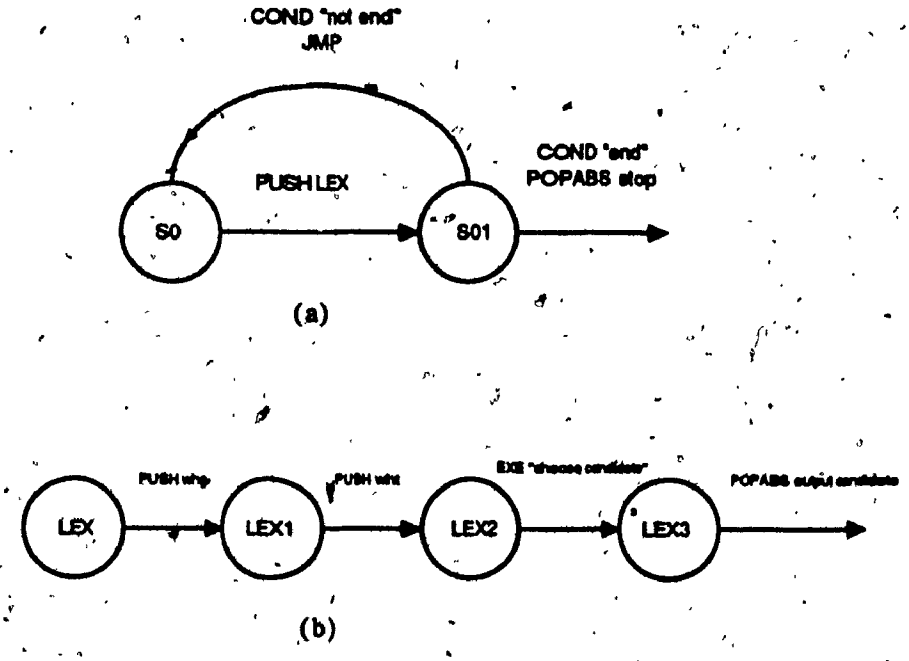


Fig. 6.1 (a) Top level of a PN for the recognition of a lexicon using (b) the "hypothesize-and-test" paradigm

The sub-network LEX, represented in Fig. 6.1b, is based on the problem solving paradigm known as hypothesize-and-test. Although this paradigm is applied here to a lexicon of isolated letters and digits it can be applied to any lexicon of any size. This paradigm can also be applied in continuous speech by starting the generation of word hypotheses at each AS detected by the data-driven segmentation process. The first arc of LEX generates a class of word hypotheses while the second arc executes word hypothesis test.

The test arc is followed by an EXE arc whose associated action chooses the best scored candidate in the best scored class. Another action associated to the last arc outputs

the selected candidate.

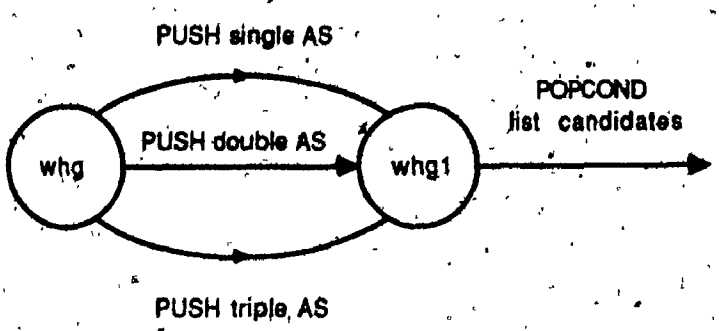


Fig. 6.2 Subnetwork for word-hypothesis generation

The word-hypothesis-generation (whg) is subdivided into three sub-networks depending on the number of ASs detected per word (see Fig. 6.2). A sub-network, characteristic of the words with a single AS, is shown in Fig. 6.3. It contains arcs with associated PUSH actions. These actions invoke sub-networks corresponding to types of AS. Types are built using the following phonetic features:

- vocalic (vw)
- fricative consonant (fr)
- plosive consonant (pl)
- sonorant consonant (sn)
- nonsonorant consonant (ns)

The feature *nonsonorant* represents fricative or plosive consonants. Each AS type that is the argument of a "PUSH" in Fig. 6.3, is represented by a Hidden Markov Model (HMM) that generates sequences of PACs. An example of the Markov Source for "fricative-vocalic" (fr-vw) is shown in Fig. 6.4. The probabilities of the HMMs of PAC symbols is learned using the Forward-Backward (FB) Algorithm [100] on strings recognized as belonging to a "fricative-vocalic" segment with an algorithm described in [20]. The "fricative-vocalic" HMM captures statistics of descriptions of this segment type based only on acoustic evidence. When an AS is analyzed, all the HMMs corresponding to segment types shown in Fig. 6.3 are executed on the string of PACs that

describes the suprasegmental properties of that AS. The segment type for which the a-priori probability of the PAC string is the highest, is selected by the function "sel-type" associated to the POBABS arc in Fig. 6.3. This "sel-type" will condition the activation of further sub-networks. Its identity and associated a-priori probability are stored into the PNs working memory and propagated through the word-hypothesis-test (wht) network.

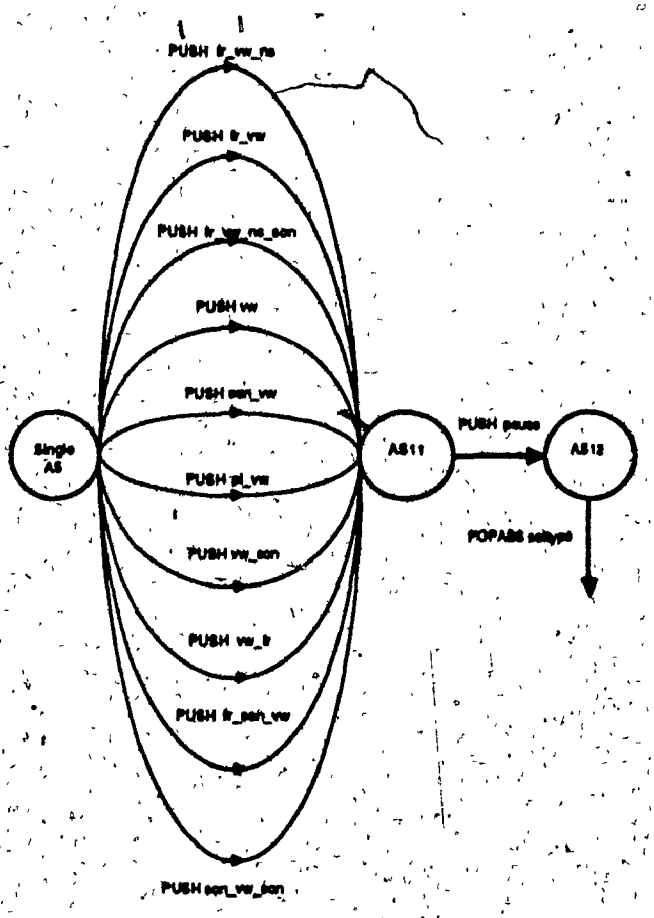


Fig. 6.3 Subnetwork for single AS hypotheses

Let:

$$Ptype = Pr\,(PAC\,(AS)\,/sel\text{-}type) \qquad (16)$$

be the score for a segment type. The supervisor uses Ptype for scoring priorities on arcs to be followed. The PN supervisor strategy for this application is essentially best-first. The arc following AS11 in Figure 6.3 is associated with a PUSH action to a sub-network that returns the probability of having a PAUSE after the just analyzed AS.

Ptype and,

$$P_{pause} = Pr((\text{signal segment following AS}) \mid \text{PAUSE}) \quad (17)$$

are multiplied and if their product is close to zero, then the "single AS" sub-network is ignored by the supervisor.

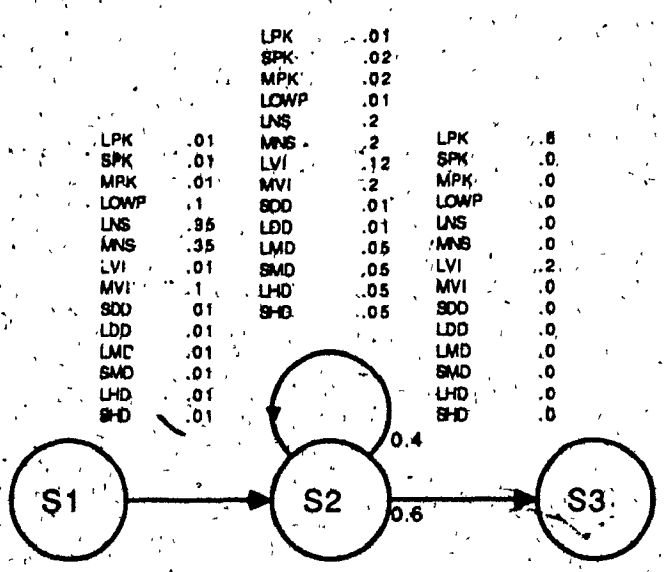


Fig. 6.4 Example of a Markov Source for "fricative-vocalic" (fr-vw)

Each AS type in Fig. 6.3. is associated with a table with the list of words that could have generated such a type with their probabilities.

For the sake of simplicity it is assumed that all words have equal a-priori probability in this application.

Word candidates for the class "fricative-vowel" are:

{ C,Z,G,J,V,Z,4 }

Other cases have to be added, which correspond to word candidates that can be erroneously classified in the class "fricative-vowel":

{ K,Q,T,3,5,0 }

Although a best-first strategy has been chosen for hypothesizing words, other strategies are available as "pruning methods" of the system.

Sub-networks similar to the one in Fig. 6.3 are available for words with 2 and 3 ASs.

3 ASs.

Each acoustic segment involved in a set of word hypotheses has been "recognized" as belonging to a segment type.

The knowledge of the segment type drives the segmentation of each AS into "head" "vocalic part" and "tail". If a word has more than one AS, then, for each AS, the "recognized" segment type is known. Notice that the knowledge of a segment type is used by the wht PN. Such a knowledge may constrain the set of candidates considered by wht but it does not necessarily restrict them to the letters belonging to the recognized segment type. For example, the letter 'T' may be verified even if the segment type "recognized" is "fricative-vocalic" if instances of 'T' have been observed before in this class. The sub-network wht has as many sub-networks as the number of acoustic segments of the word hypotheses to be verified. The sub-network wht1 for a single AS is shown in Fig. 6.5. The AS head is analyzed by attached procedures (actions) performing an Elaboration-Decision (ED) paradigm. These types of procedures are called ED-actions. ED-actions perform variable-depth analysis on subsegments of AS. The details of ED-actions were described in section 2.4. There are three possible ED-actions for the head of an AS, namely:

- plosive head
- fricative (including afficate) head
- sonorant head.

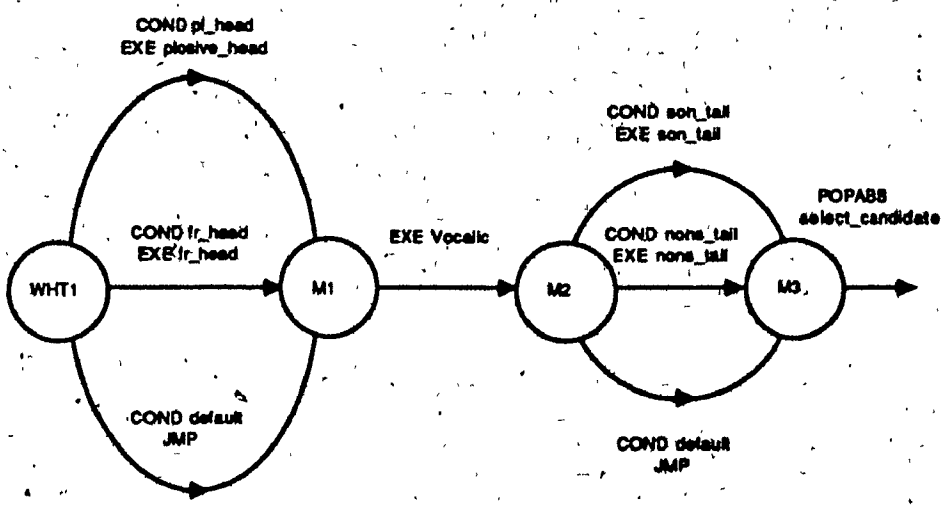


Fig. 6.5 Subnetwork for hypothesis test

The choice of the ED action is made by disjoint conditions associated to arcs. These conditions are predicates on the segtype associated to the AS. State M1 in Fig. 6.5 will be reached by only one arc. As it will be seen later, the ED action for the AS head identifies the AS head as an acoustic subsegment and extracts some acoustic properties. Using techniques partially described elsewhere [68], hypotheses about the place and manner of articulation for the speech unit in the head subsegment are generated and scored by the following a-priori probability:

$$Prh = Pr(\text{data}(h) / \text{place} \cap \text{manner}) \quad (18)$$

place is a variable that takes values in the following set:

$$PLset = \{\text{labial, front, central, back}\} \quad (19)$$

manner is a variable that take values in the following set:

$$MNset = \{\text{high-vowel, low-vowel, oral, nasal, unvoiced, voiced-nonsonorant}\}. \quad (20)$$

The description data(h) contains acoustic properties the system knowledge considers worth extracting given the suprasegmental characteristics of the head subsegment. These properties can be broad-band spectral energies for a fricative head or narrow-band spectral lines for a sonorant head. PACs are also used in data(h). After state M1, the ED-action "vocalic" is executed. It segments the vocalic part of AS into stationary and transient units.

Let,

$$v_1\, v_2 \ldots v_x \ldots v_X$$

be such subsegments. For each segment v_x spectral lines are considered as data and a-priori probabilities about place and manner of articulation are obtained by HMMs of spectral lines in the segment v_x as explained in chapter 4. For each subsegment and for each consistent *place-manner* pair, the following probability is computed.

$$Pr(v_x) = Pr(data(v_x)/place \cap manner) \quad (21)$$

Also for v_x, more knowledge pertaining to the vocalic segment is calculated and the evidence of properties are also associated in $Pr(v_x)$ as explained in chapter 3 and chapter 4. From state M2 to state M3, ED-actions for the tail of AS are executed similar to those used for the head. A probability,

$$Prt = Pr(\text{data (t) /place} \cap \text{manner}) \quad (22)$$

scores the hypotheses of the tail subsegment. The data extracted in the head, the subsegments of the vocalic part and the tail can be assumed to be independent. Let data(m) be the data used for a monosyllabic hypotheses, then:

$$data(m) = data(h) \cup data(v_1) \cup \ldots \cup data(v_x) \cup \ldots \cup data(v_x) \cup data(t) \quad (23)$$

The "select" action associated with the POPABS arc knows for every monosyllabic hypothesis the segtype and the place and manner of articulation of each phoneme of the hypotheses. As all the required probabilities are propagated through the PN, the best probability for each candidate hypothesis can be computed as follows:

$$Pr(data(m)/hyp) = Pr(data(m) / sequence(place(i) \cap manner(i))) \quad (24)$$

where *sequence* indicates the sequence

$$P_{h1}, P_{h2}, \ldots, P_{hi}, \ldots P_{hl} \quad (25)$$

of the hypothesis phonemes, place (i) and manner (i) represent respectively the place and manner of articulation of phoneme P_{hi}.

Duration statistics for the Speech Units involved in each hypothesis can be collected and used in the "select action". As probabilities for places and manners of articulation are computed in well delimited time intervals, durations of these intervals can be considered as additional data.

The segmenter that produces ASs may undersegment and, very rarely, oversegment. In both cases, hypothesis verification sub-networks are considered with many or no vocalic segments. A maximum of two and a minimum of zero segments are allowed for both head and tail.

6.2 The Test Data

A corpus consisting of 400 pronunciations of the vocabulary defined in Table 2 was used for evaluating the ASR model developed based on the theory presented in this work. The corpus was obtained by asking 100 (50 male and 50 female) speakers to pronounce four times the entire vocabulary.

Speakers were mostly University students and instructors with different mother tongues. They were all asked to speak in English.

A Computer program generated random sequences of 5 letters or digits. Each speaker was asked to pronounce each sequence presented to him/her with a little pause

between each letter or digit. Data was acquired with a Hewlett Packard Special Purpose Workstation HP 9000-236 kindly donated by HP.

Signals were sampled at 10kHz over 12 bits. The signal was windowed by a 23msecs. Hamming window and a 128 points Fast Fourier Transformation (FFT) was computed every 10msecs. by a TMS 320.

The rest of the processing was carried out on a VAX 8600 although a distributed version of the recognition system following a paradigm proposed in [20] is being implemented. This new version uses a TMS 320 and the two processors of the HP 9000-236 and HP 9000-320 Workstations.

Learning was done incrementally as proposed in [68]. A first version of the system knowledge was set up using 80 pronounciations of the vocabulary from 20 speakers.

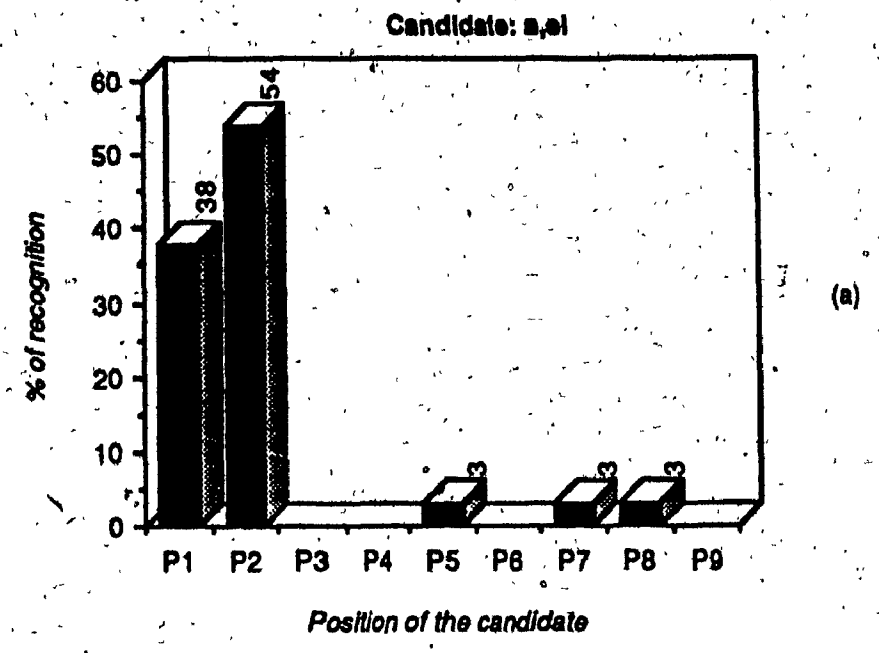
Another 20 speakers (10 male, 10 female) were used for testing. A sample for each word of the 36-word vocabulary was used for each speaker resulting in 20 samples for each vocabulary word.

6.3 Experimental Results

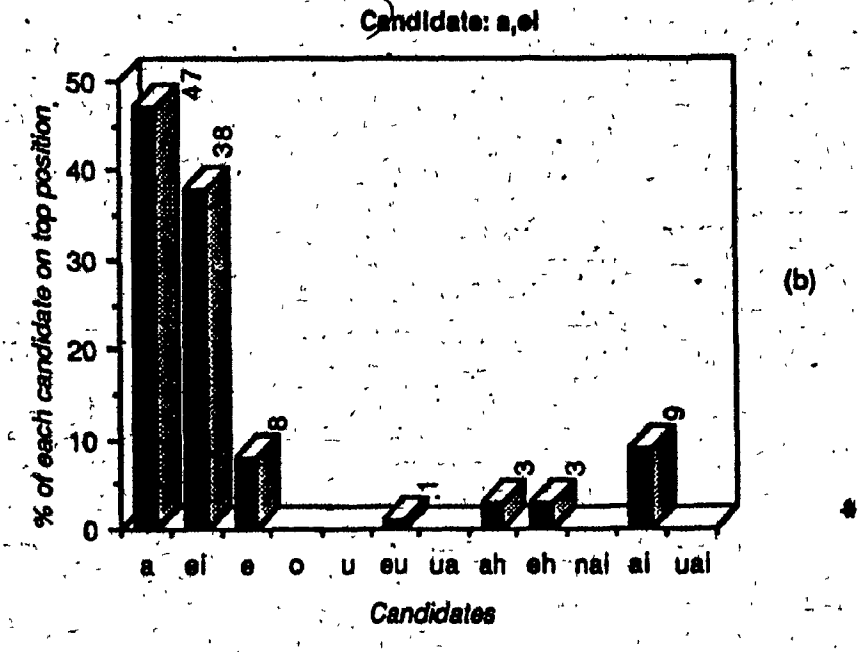
Several experimental results are reported in this section. The letter-digit database is used to evaluate the performance of the SPA sub-network as well as to evaluate the overall performance of the whole PN based recognition system. In the following sections the results obtained from both of the systems are summarized.

6.3.1 Performance of the SPA Sub-network

The recognition of vowels and diphthongs represented as speech units belonging to the set {a, e, ei, ai, ah, eh, o, u, eu, ua, nai, uai} are shown in Figures 6.6 through 6.15. For each candidate in the above set, two graphs are given: (a) the position, P1, P2, . . ., P9 of the right candidate, where, P1 is the top position and P9 is the last position and (b) the plot showing the percentage of each competing candidate being on the top position, P1. It is important to notice that the right candidate being in the 2^{nd}, 3^{rd}, or even in the 4^{th} position is considered to be acceptable since scores from different sources are considered by the PN to make the final word hypothesis. This way, a possible higher score returned by another sub-network may help in correctly recognizing the word. Various illustrations of such situations are given in the next section. Notice that the

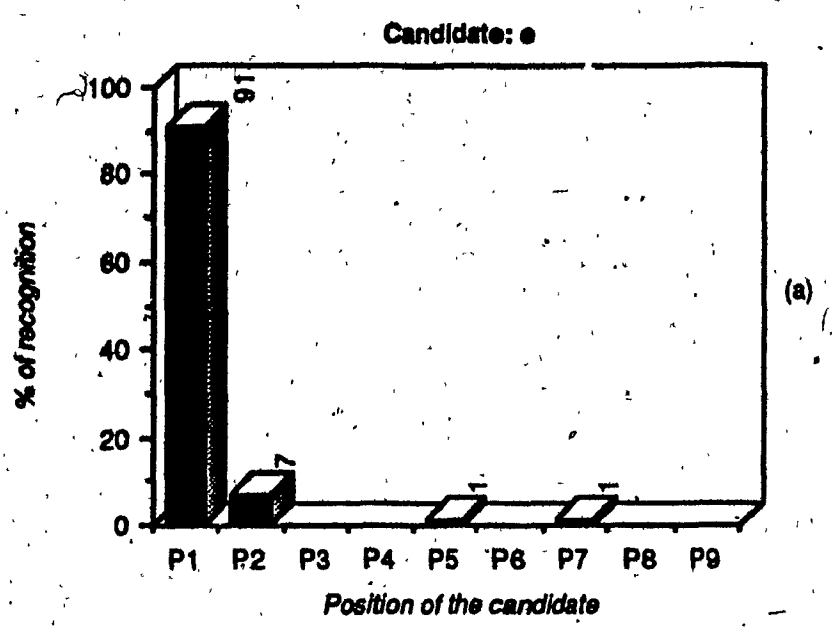


(a)



(b)

Fig. 6.6 Correct identification of candidates a,ei (a) shows percent recognized on top position, P1 through last position, P9 and (b) shows other competing candidates.



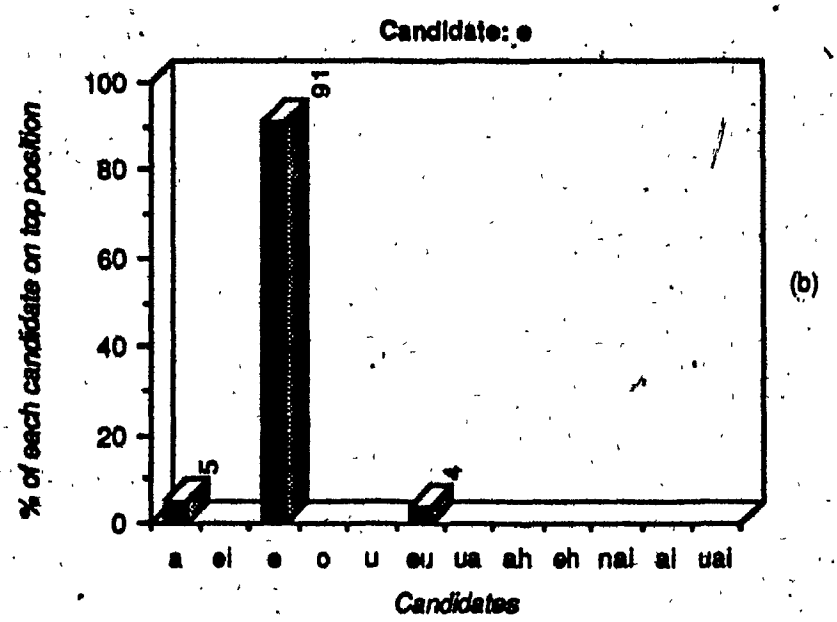
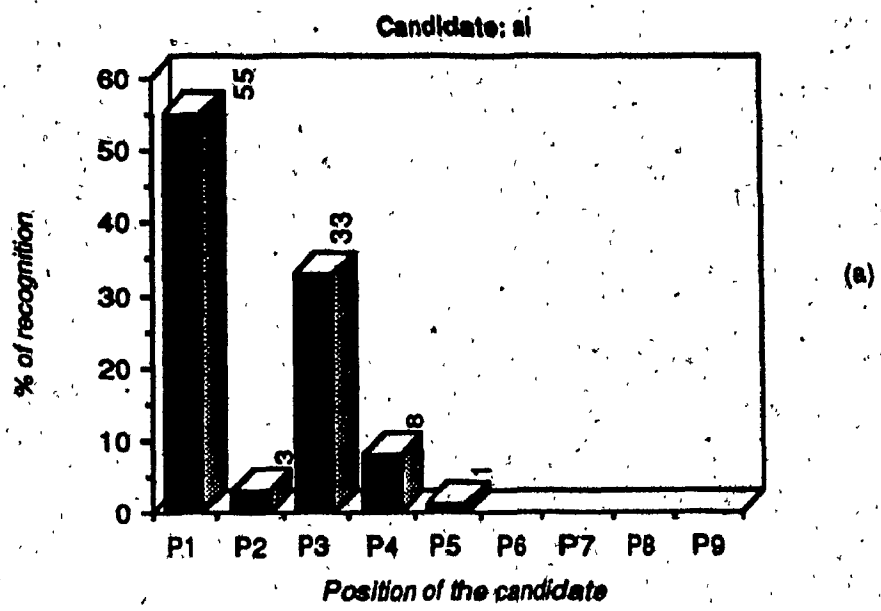


Fig. 6.7 Correct identification of candidate e (a) shows percent recognized on top position, P1 through last position, P9 and (b) shows other competing candidates.



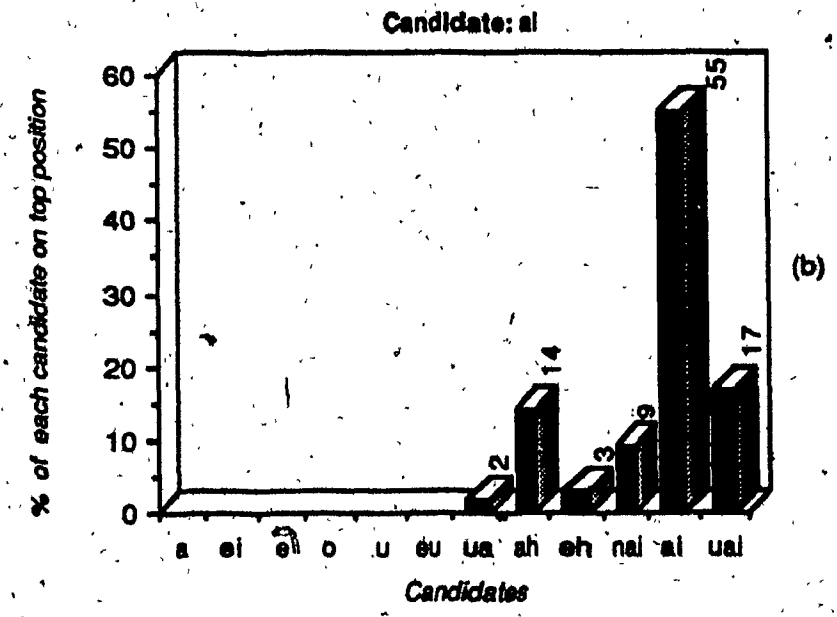
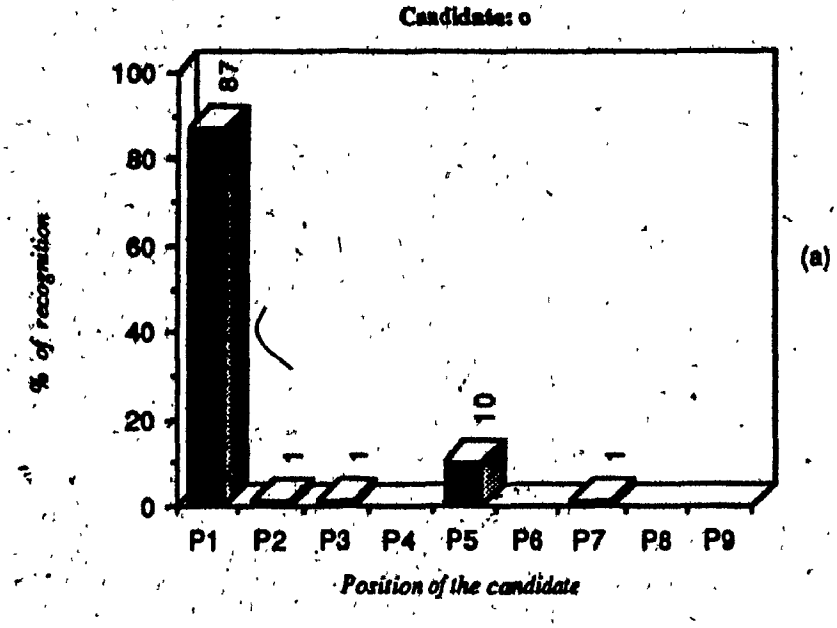


Fig. 6.8 Correct identification of candidate ai (a) shows percent recognized on top position, P1 through last position, P9 and (b) shows other competing candidates.



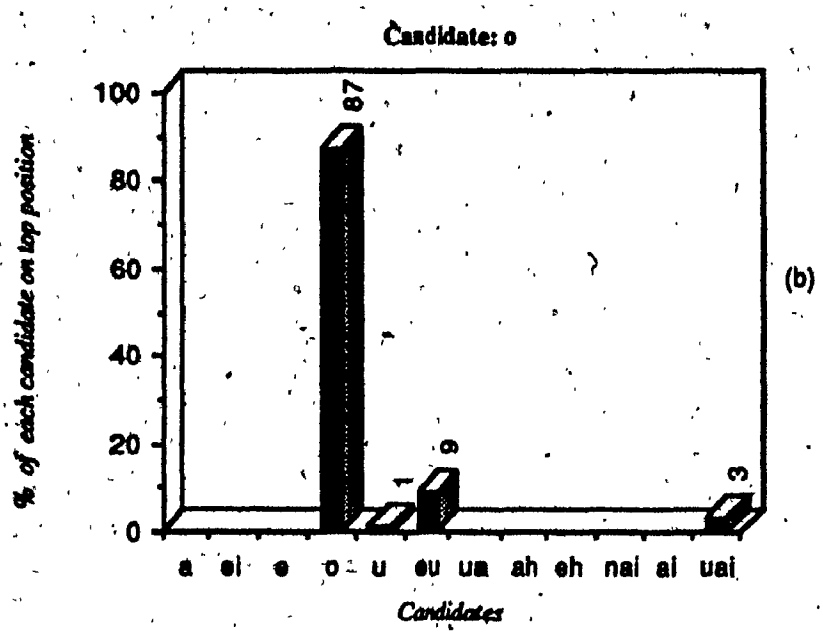
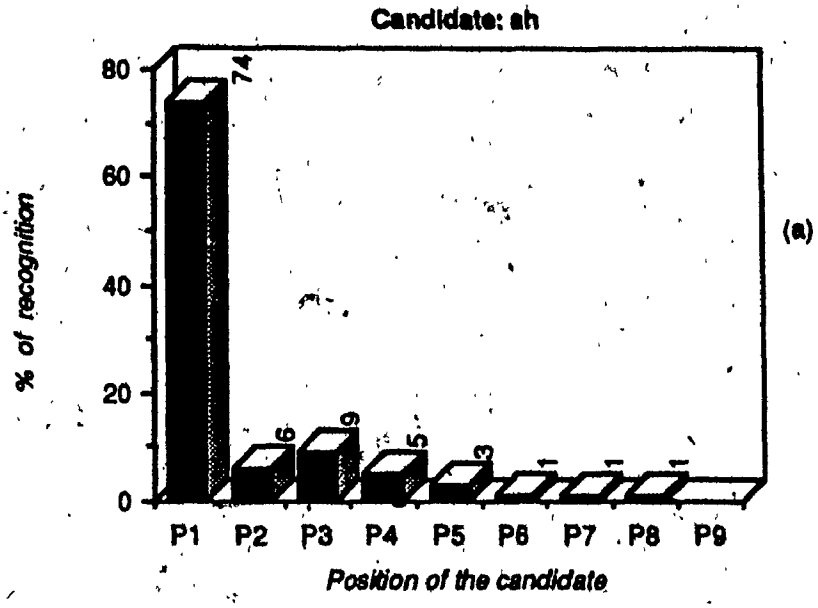


Fig. 6.9 Correct identification of candidate o (a) shows percent recognized on top position, P1 through last position, P9 and (b) shows other competing candidates.



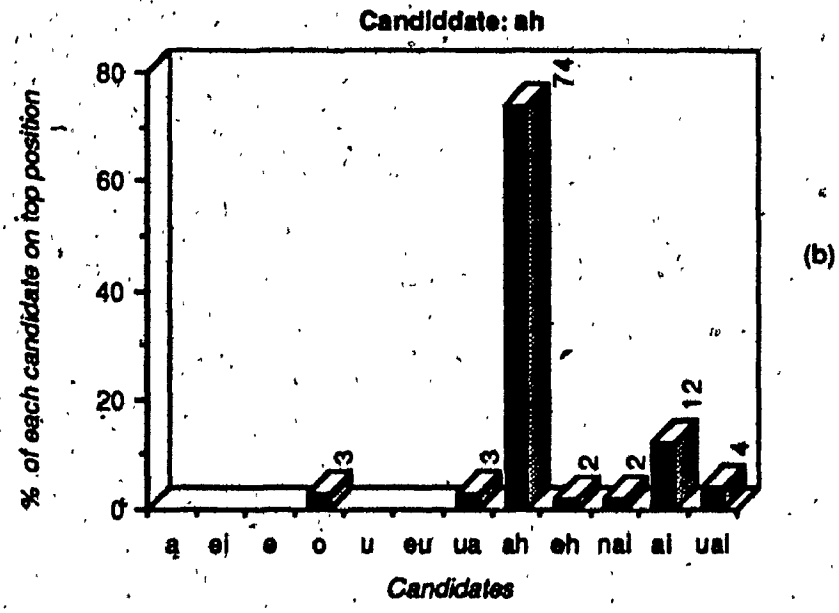
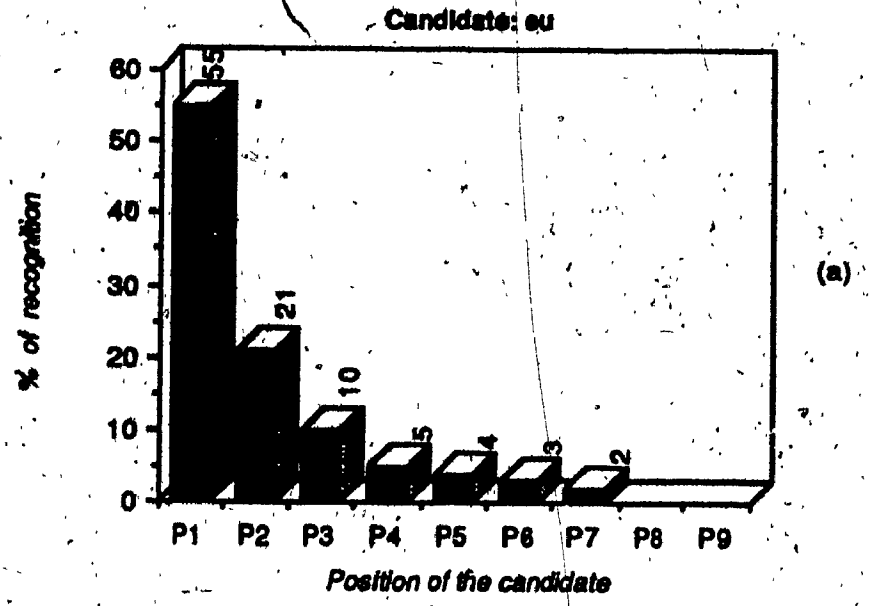


Fig. 6.10 Correct identification of candidate ah (a) shows percent recognized on top position, P1 through last position, P9 and (b) shows other competing candidates.



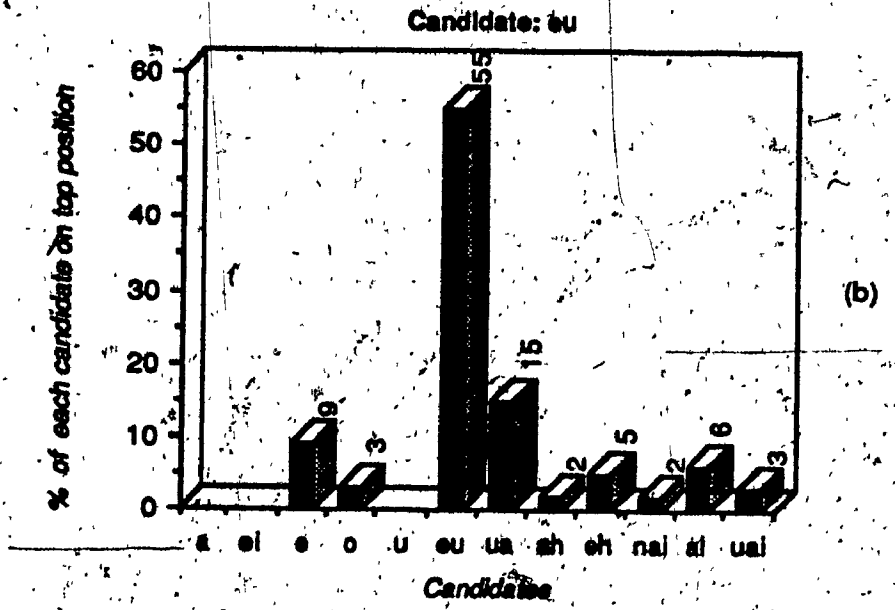


Fig. 6:11 Correct identification of candidates u,eu (a) shows percent recognized on top position, P1 through last position, P9 and (b) shows other competing candidates.

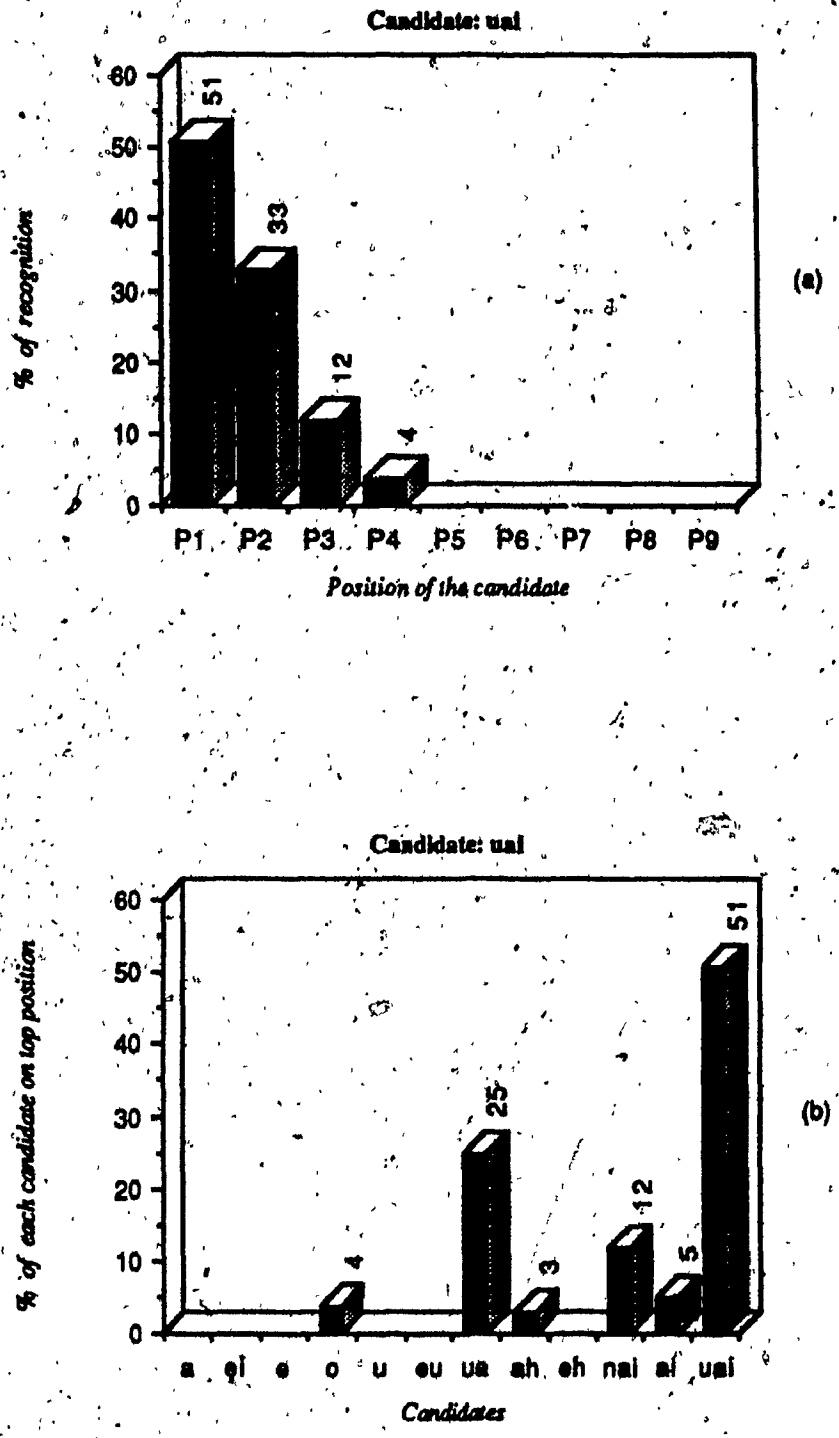


Fig. 6.12 Correct identification of candidate uai (a) shows percent recognized on top position, P1 through last position, P9 and (b) shows other competing candidates.

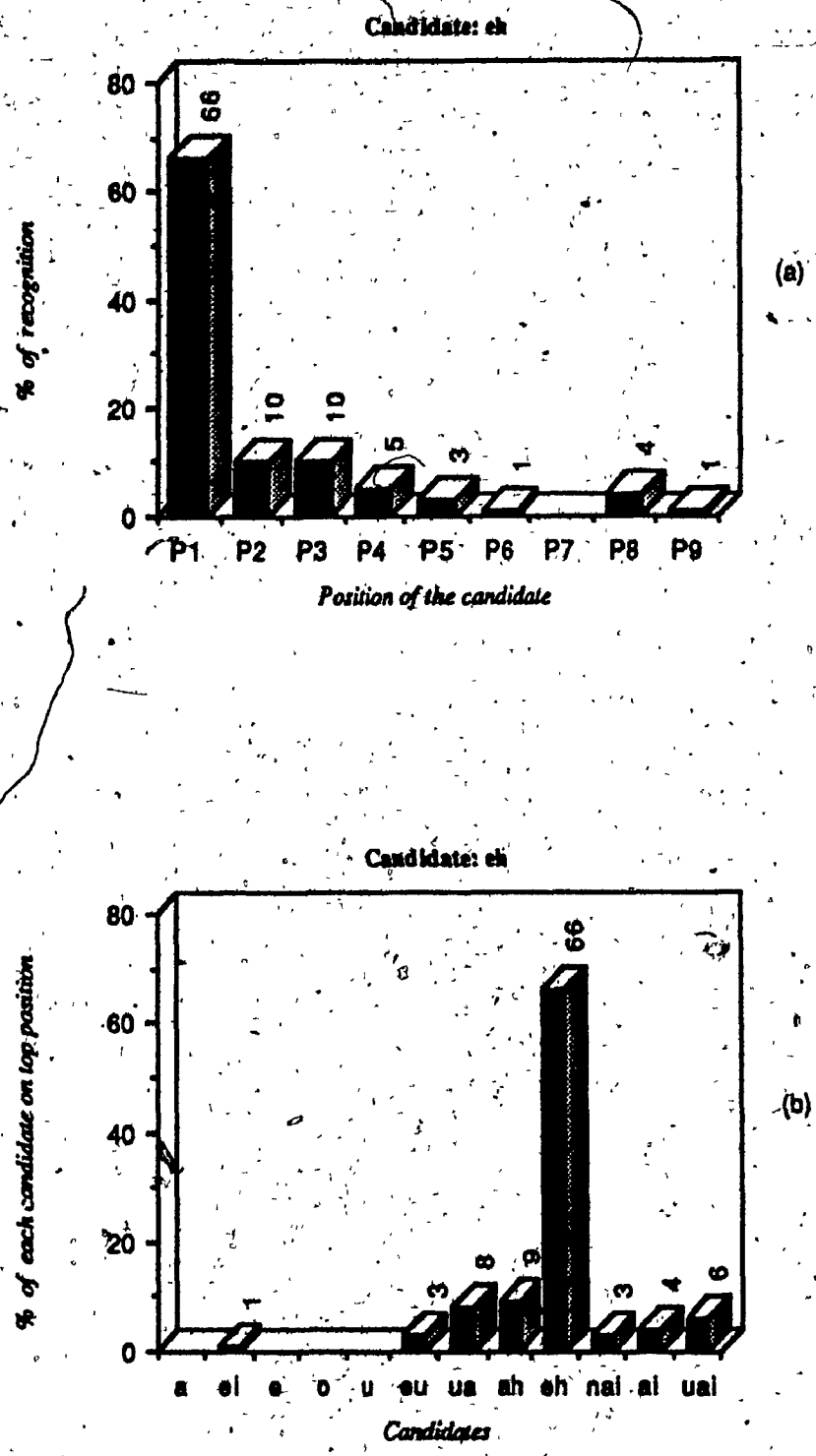
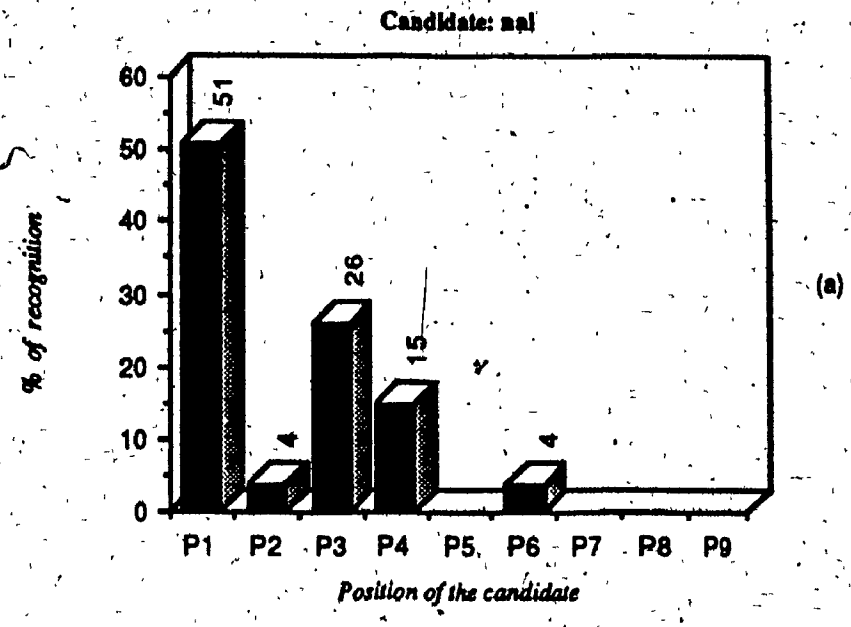


Fig. 6.13 Correct identification of candidate eh (a) shows percent recognized on top position, P1 through last position, P9 and (b) shows other competing candidates.



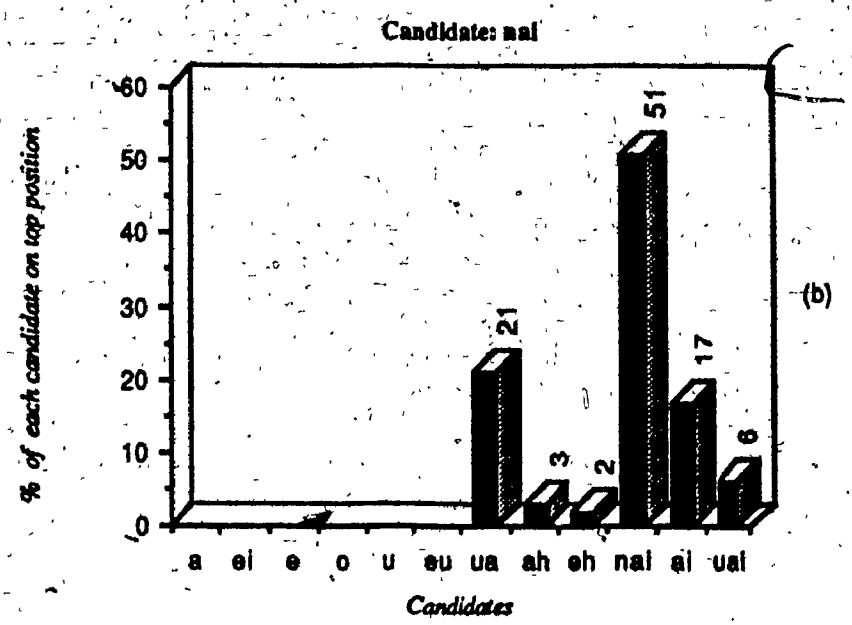
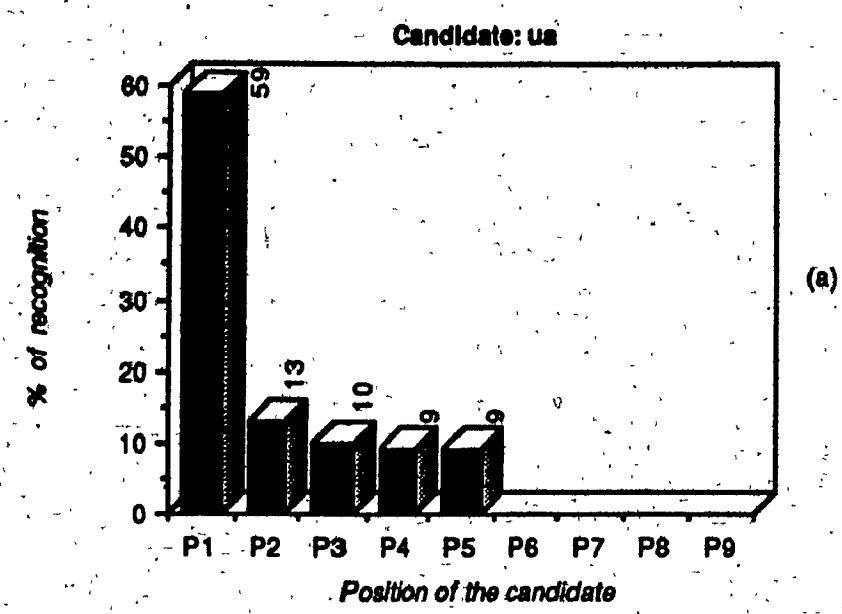
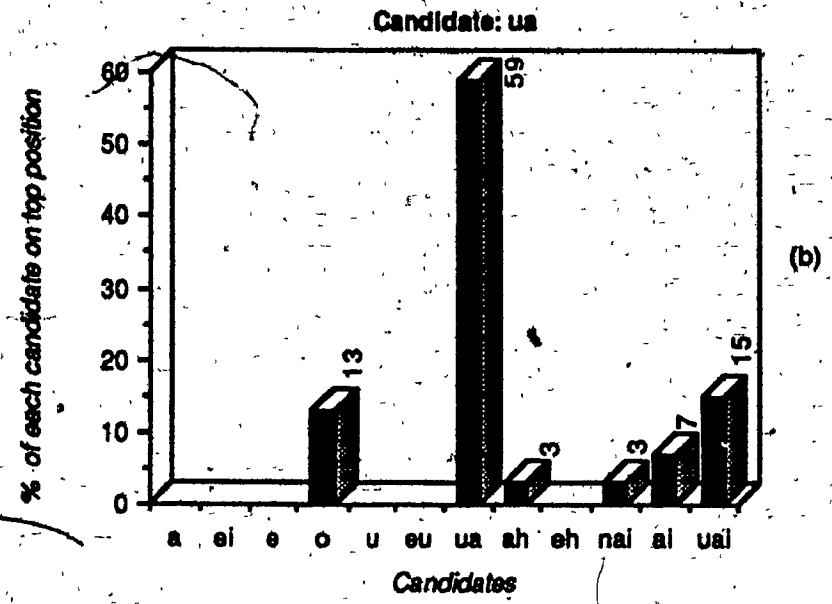


Fig. 6.14 Correct identification of candidate nai (a) shows percent recognized on top position, P1 through last position, P9 and (b) shows other competing candidates.



(a)



(b)

Fig. 6.15 Correct identification of candidate ua (a) shows percent recognized on top position, P1 through last position, P9 and (b) shows other competing candidates.

speech units {a,ei} are considered as same, even though they are represented differently. Therefore, in the recognition graphs, there is only one graph corresponding to both the candidates. The candidates {u,eu} are also considered the same way.

6.3.2 Performance of the PN System

The Procedural Network incorporates hypotheses from all other sub-network in order to make the final word hypothesis. Table 6.2 shows the recognition of certain letters and digits based on the scores reported by other sub-network. Note that in Table 6.2, the head-analyzer sends back scores for a set of probable candidates even if there is no "head" is present. Case (a) in Table 6.2 shows an example where both head-analyzer and the SPA sends top scores for the candidate "9" in which case the final recognition is obvious. In case (b), the head-analyzer returns top score for "H-HEAD", SPA returns top score for "EH", and the fricative-tail-analyzer returns top score for "X-TAIL" which eventually leads "X" to be the final word candidate. More examples about similar situations are given in Table 6.2.

The results shown in Table 6.2 illustrates that it is not absolutely necessary for all sub-network to return top scores for the right candidate (it is still desiarable but not always possible). However, if most of scores are poor then the system would fail to make the correct decision. From experimental observations, it has been found that when all of the sub-network fail to return a reasonably good score, then there is a root problem mostly associated with the speech signal. The problems may have occured during speech acquisition, digitization, or may caused from some other sources which leave the speech signal usually incomplete.

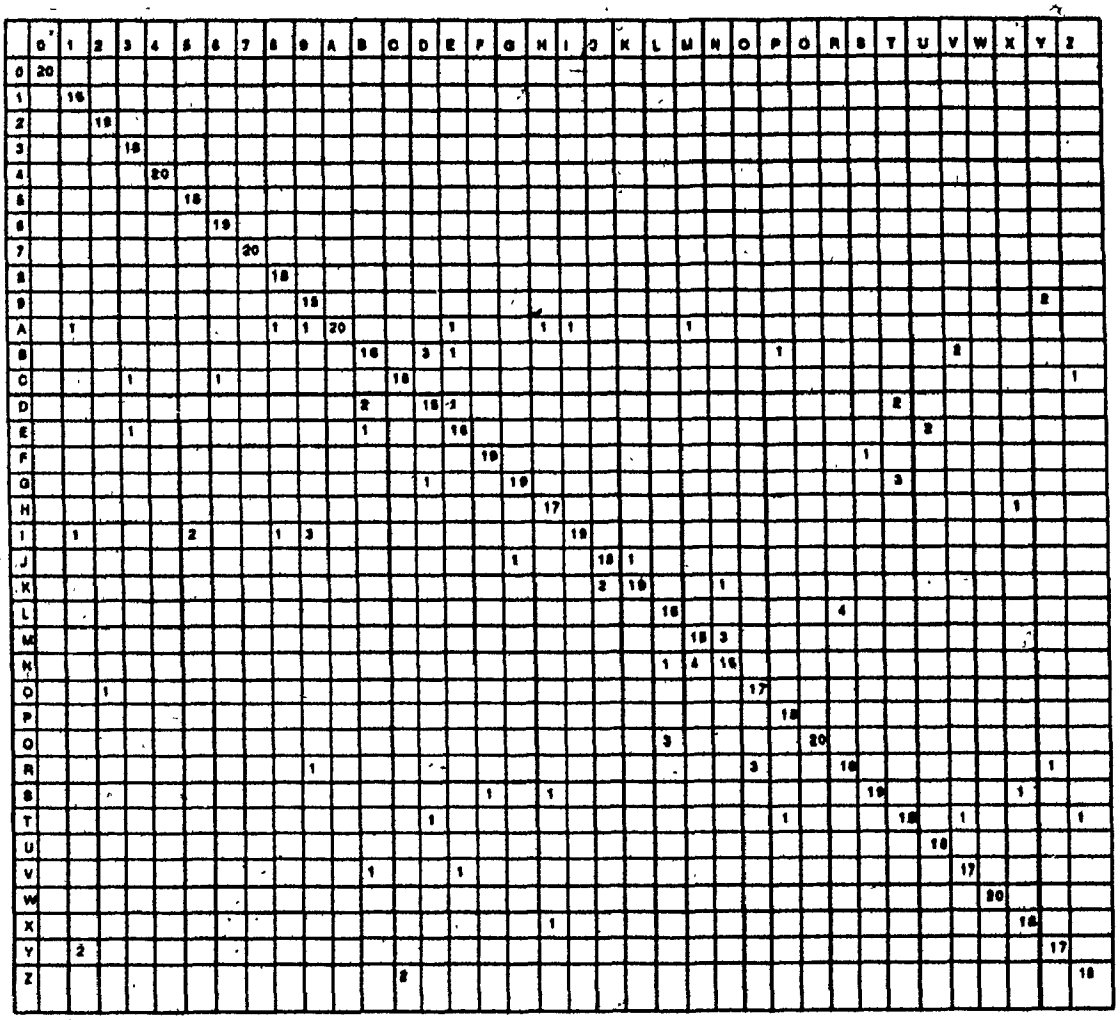
An overall recognition of system for the 36 word vocabulary is given in Table 6.3. Column headings represent pronounced words, row headings represent recognized words. An overall recognition rate of approximately 90% was achieved.

Errors are expected to become lower as incremental learning proceeds in order to adapt system knowledge and statistics to new speakers. Nevertheless, as the probability of having the right candidate in the top-3 positions is very high, this system is suitable for applications in which coded information (like file names) has to be accessed and enough redundancy is provided in the coding.

Table 6.2 Sample output from Procedural Network

	Head Analysis		Vowel Analysis		Tail Analysis		Final Word Hypothesis	
	Candidates	Score	Candidates	Score	Candidates	Score	Candidates	Score
(a) Word spoken: "9"	Nine-Head	0.1222510	NAI	0.175409	No-Tail		WRD-9	0.0214439
	Y-Head	0.0695653	EU	0.165429			WRD-Y	0.0122024
	B-Head	0.0677044	UAI	0.163469			WRD-1	0.0118247
	N-Head	0.0677023	UA	0.155409			WRD-U	0.0118075
	One-Head	0.0674121	AI	0.155143				
	U-Head	0.0673142	O	0.143222				
(b) Word spoken: "X"	H-Head	0.5778050	EH	0.231004	X-Tail	0.830045	WRD-X	0.110790
	X-Head	0.4221950	AH	0.231004	CH-Tail	0.169955		
			AI	0.167448				
			UAI	0.160725				
			NAI	0.104909				
			UA	0.104909				
(c) Word spoken: "2"	Two-Head	0.6451610	O	0.248588	No-Tail		WRD-2	0.160379
	P-Head	0.3548390	U	0.230333				
			EU	0.143961				
			NAI	0.125706				
			UAI	0.125706				
			UA	0.125706				
(d) Word spoken: "H"	X-Head	0.5997240	EI	0.277351	X-Tail	0.591760	WRD-H	0.0679042
	H-Head	0.4002760	AI	0.228686	CH-Tail	0.408240		
			NAI	0.133553				
			UAI	0.133553				
			UA	0.133553				
			A	0.093304				
(e) Word spoken: "K"	K-Head	0.4105370	E	0.296718	No-Tail		WRD-K	0.0881365
	Zero-Head	0.3636670	EI	0.214686			WRD-T	0.0669978
	T-Head	0.2257960	A	0.166567			WRD-O	0.0390371
			NAI	0.107343				
			EU	0.107343				
			UAI	0.107343				

Table 6.3 Recognition table for the 36 word vocabulary



Chapter 7

Discussion

In this chapter we discuss the underlying assumptions and contributions made during this research work.

Several novel ideas are presented for solving various tasks pertinent to Automatic Speech Recognition. The knowledge based Procedural Network using variable depth analysis approach is used as the main recognition model. Some of the advantages of using PNs as recognition models are discussed.

The application of the Biological Vision concepts for interpreting speech spectrograms is the major contribution made in this thesis work. In conjunction with the vision approach, the use of frequency based Continuous Parameter Markov Models for identifying vowels in quasi-stationary regions in the speech signal is proposed. A brief discussion on these two concepts will be considered in the following discussion.

Reasons for using acoustic segments as initial input to the Speech Pattern Analyzer (SPA) are discussed in detail. Internal segmentation, a technique used to capture contextual variations within acoustic segments is also considered in the discussion. In addition, the scoring technique based on Dynamic Weight Adjustment (DWA) is discussed in detail.

7.1 The Procedural Network Approach.

The Procedural Network (PN) model is used to integrate cognitive and information-theoretic approaches. Since the cognitive approach attempts to infer maximum speech knowledge and the information-theoretic approach can learn statistically the changing behavior of speech, integration of both would be the ideal solution for achieving speaker-independency.

The Procedural Network approach is also appealing because it can be expanded

into connected speech using large vocabularies. Unlike other classical recognition systems, the knowledge based procedural network system incorporates techniques of various other recognition systems in one. Techniques such as, the hierarchical organization, use of independent knowledge sources, and above all, a plan based recognition strategy are all incorporated in this model.

Use of speech knowledge in speech recognition systems has received attention ever since the ARPA project. By incorporating speech knowledge as well as knowledge about the problem domain the errors generated by the recognition systems become more evident and justifiable. This way, if the source of error is clear, more knowledge can be associated to correct the errors. The Procedural Network approach allows easy insertion of sub-networks where each sub-network can act as a knowledge extractor. Therefore, knowledge integration is basically an inherent property of the PNs.

Knowledge extraction and representation can also be formulated in different ways like, pure statistical, stochastic, or any other techniques used in the Artificial Intelligence area. The scores returned from sub-networks can be probabilistic or simple normalized measures. PN can associate these scores, which are observed in different time frames and context in its hypothesis paradigm.

The last but most important task of PN, as well as its sub-network is to achieve speaker independency. Speaker independency is achieved by considering, for each speech segment in a given word candidate, all the possible sound types in a head-vowel-tail organization. The head and tail may not always be present for every word but the vowel parts will be present.

Inter- and intra-speaker variations cause properties in head, tail, or vowel regions to be weak, distorted , or even absent by unexpected noise caused by aspiration, coarticulation and environmental noises. The evaluation of head-vowel-tail components is done by different sub-networks, each sub-network returning a score for its part, based on its knowledge. The scores returned by different sub-networks may not necessarily correspond to the same word candidate. For example, for letter "k", the plosive-head analyzer may return a confusion set {t,k} with score for "t" greater than score for "k". However, the vowel-analyzer may return a confusion set {a, ei, e} as vowel or diphthong with a and ei having high scores. Since e for "t" is in the third position, the PN would rule out the candidate "t" from the confusion set as the top candidate. This way, by integrating knowledge from different sources, the PN tries to achieve speaker-independency.

The reason why plosive-head analyzer returns a lower score for "k" may be

because the sub-network which performed the task did not have sufficient knowledge or the speech for the head region was distorted and the cue to recognize the k-head was not detected in the signal. If the problem was due to lack of knowledge, new knowledge could be incorporated into the sub-networks. But if it was a problem of distortion, the sub-network simply has to return a score based on available knowledge applied on such situation. Whatever the reason, the fact that the sub-network returned a score for all likely candidates will enable the recognition system to make the correct hypothesis.

The knowledge is associated with different sound classes and not with the candidates in the vocabulary set. Since knowledge about a particular sound class must be finite, the process of upgrading knowledge will not be necessary after observing several variations of a sound class. The choice of Procedural Network as the recognition model is proven to be a good approach for achieving speaker independency.

7.2 The Biological Vision Approach

A robust vowel-diphthong detector is an integral part of any ASR system. The problem of vowel recognition is not new and extensive work has been done in the past in this area. Unfortunately, most of the vowel recognition algorithms use parameters extracted by various signal processing algorithms. These algorithms incorporate little speech knowledge and are not capable of adapting to variations in the speech signal.

This thesis work proposes a technique to capture speech knowledge which is available in spectrograms and treat it as a scene. A simple pattern analysis technique applied to these patterns reveals significant properties which are relevant to transitions of vocal tract as well as being speaker independent in nature. This process is labeled under Biological Vision mainly because of the following reasons:

a) Unlike classical vision problems handled by machines, there is no need for any sort of complicated transformations or rotations in order to interpret these speech patterns.

b) Biological vision system uses a global recognition strategy by considering the image as a "whole". The recognition processor, the brain, uses symbols and symbolic relationships in image for image interpretation. Also, the knowledge base consists of symbols as well as symbolic relationship of objects in its long term memory.

In order to give the machine a similar capability as that of biological vision systems, the pattern of a speech spectrogram is described as a morphology of symbols and symbolic relationships. The morphological description is carried out at various levels, and at each level, the lines in the pattern are described as symbols or as a relation between symbols.

As new variations appear in the pattern for similar objects, the symbolic knowledge base is updated. The symbols in the knowledge base are speech properties that describe a scene as well as the the changes in the scene.

The allowable degradations for any given scene, (a scene corresponding to a particular vowel or diphthong; each vowel or diphthong having a unique or similar scene appearance) are finite and caused by speaker variations and noise in the signal. The knowledge about a scene and its variations can be learned statistically or be represented by rules.

The idea of treating speech spectrograms as patterns, describing them in a symbolic way, and learning the variations statistically achieves the expected goals, namely, speaker-independency and robust recognition of vowels and diphthongs in any given context.

7.3 Primary Acoustic Segments as Input to SPA

Primary acoustic cues described in Table 2.1 are the input to SPA. Segment type representations are always better than time-frame representations, because a segment can span over a region of speech and be characterized over time. Several strong cues of speech sound are distributed over time. SPA needs to be applied only on stationary or quasi-stationary speech regions. Note that these regions are not present in acoustic segments belonging to the acoustic cue set [LDD, LNS, MNS, SNS]. Since pattern analysis and segmentation is time costly, there is no need for the above segments to be included. Therefore, only those acoustic segments exhibiting stationary properties need to be considered for SPA.

7.4 SPA as a Pre-Processor

SPA can also be considered as an acoustic-phonetic level pre-processor. The task of a pre-processor is to rule out unlikely candidates prior to next level processing.

Pre-processing by SPA is carried out at various levels. At the level of pattern creation and pre-processing, if the given acoustic segment is highly unstable the whole segment, or part of the segment, can be rejected or can be described as a non-sonorant region. If an acoustic segment is classified as non-sonorant, then further processing of that particular acoustic segment is suspended. Examples of this are the NSPH and NSPT properties. A non-sonorant region may appear at the beginning, within, or at the end of a given acoustic segment sequence. If it appears at the beginning or at the end, then it could represent a problem caused by the acoustic segmenter. If within, then it could be due to noisy transitions or the existence of a non-sonorant region between two sonorant regions. For example the /r/ of "zero" occurs between /e/ and /o/. Identifying and removing such segments before the next level processing by SPA would reduce certain errors as well as enhancing the speed of SPA's performance. Notice that SPA is capable of analyzing only sonarant regions.

Collapsing or encoding the sequence of vowels, $v_1....v_x$ generated by CPMM is another example of pre-processing. By encoding the sequence, the number of possible combinations is greatly reduced which in turn produces a small set of pre-conditions to be used for next level processing.

Classical pre-processors filter word candidates primarily on detailed spectral information. Pre-processing based on parametric information is not robust, since pre-processing removes word candidates and the word-hypothesis may not be able to recover these errors. The decision thresholds must be lenient to avoid irremovable errors. Therefore, the information given to the pre-processor must also be robust.

The pre-processing stages which are used here are robust since several pieces of knowledge are associated with the decision thresholds. Most of the decision criteria are symbolic properties which represent time-varying properties as well as detailed decisions based upon speech knowledge.

7.5 Internal Segmentation

Acoustic segments (the input to SPA) correspond to global events which reflect gross spectral changes. Internal segmentation, used to capture intra-segmental variations, is used in this work for two main reasons: (1) to capture variations occurring within sonorant regions of the signal and (2) to eliminate or re-label non-sonorant regions.

Certain parameters extracted from internal segments are used for the

recognition of vocalic type speech units by Markov models. The parameters for the CPMM could be extracted on a frame-by-frame basis. However, this would be computationally expensive. Another disadvantage is that, even a little variation in the location of the anchor point would cause different vocalic symbols to be generated for successive frames (eg. <VB VF VF VB VC VF VF.....>). A smart pre-processor is needed at this point to differentiate between real changes and noisy oscillations. By considering several frames as a segment, non-significant changes are ignored by the internal segmentation algorithm. Fig 7.1a shows an illustration of frame based labeling and Fig. 7.1b shows a segment based labeling. Notice that, the same CPMM's are used in both cases for labeling.

To summarize, the use of vocalic labeling on internal segments allows to capture intra-segmental variations by keeping the computational overhead minimal.

7.6 The CPMM

The use of HMM in speech recognition is not new. Several systems in the past have used such the approach described in sec 4.7. However, the HMMs in this thesis are used in a different way. In classical systems, HMMs were used to recognize words.The disadvantage is that an HMM is needed for each word in the vocabulary. Most of the successful systems use fixed duration models and are applied mainly in speaker-dependent systems. Symbols for these systems were from segments of speech 10 msecs. in length, and were obtained by vector quantization, a process that is speaker-dependent and context-independent.

The proposed CPMM is used as a tool for characterizing variable length segments which are quasi-stationary. Since only three symbols, back (VB), central (VC), and front (VF), are used for labeling, the number of models needed is very small and independent of the vocabulary used. A change in vocabulary has no effect on the models.

Another advantage of this model is that the symbols to be learned, (VB,VC,VF), may be available in more than one place for a given word. For example, the word 'SEVEN' has two VFs in it. Therefore, a large number of strings for VBs, VCs, and VFs can be collected from a given number of words (if the words are carefully selected). The model chains used for training VB, VC, and VF, were originally collected from a connected-digit corpus. These models are used in isolated letters and digits without requiring retraining.

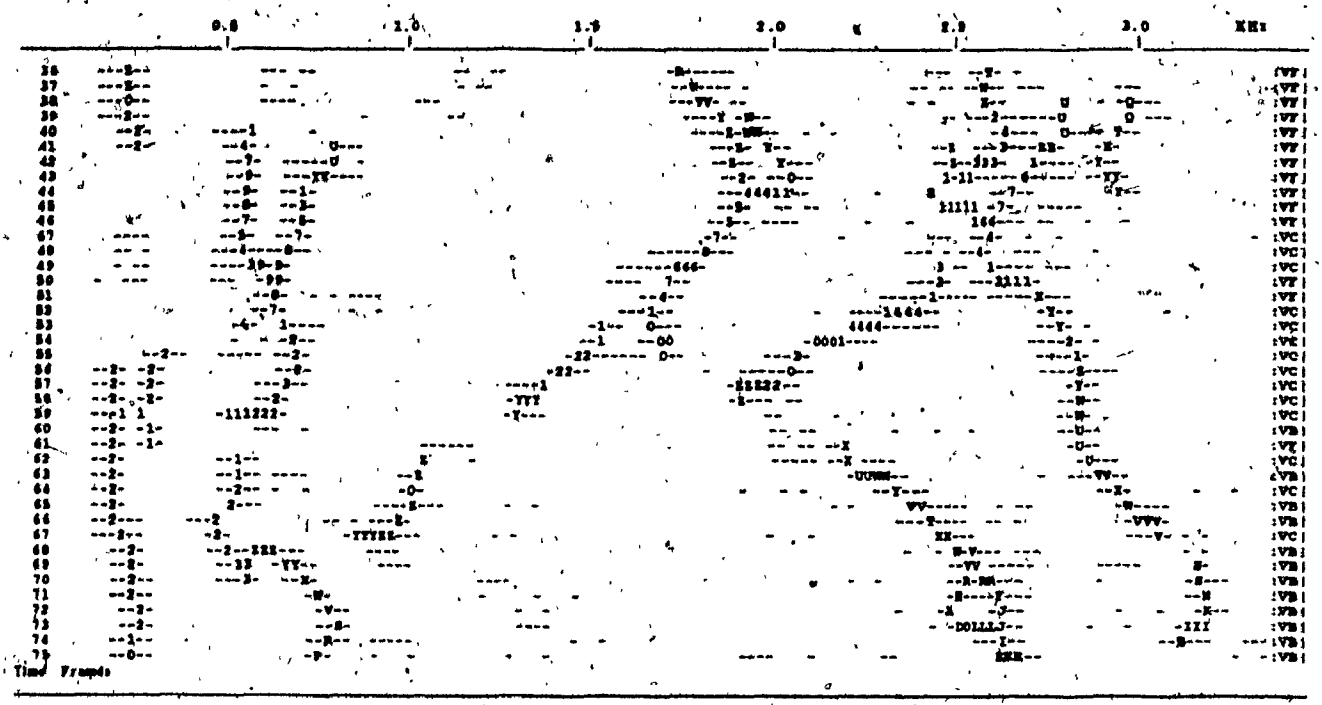


Fig. 7.1a Example of frame based labeling.

A third advantage of this model is the small number of states used. Since these models only have to recognize speech units of vowel catagories, and vowels are characterized based on first three formants (F_1, F_2, F_3), an ideal model needs no more than 10 states. In the model, (as explained in section 4.2), vowels are characterized based on locality and strength of significant and non significant lines in significant frequency regions.

CPMM is used to achieve speaker independent recognition of the place-of-articulation of vocalic regions. Since deviations of frequencies and amplitudes of spectral lines around target values reflect variations among speakers, and this can be characterized statistically, the model performed extremely well even for the speaker-independent task.

A vowel symbol (label) is given by the CPMM for each internal segment and a sequence of acoustic segments may have several internal segments. Time varying nature within the segment is obtained by concatenating the sequentially generated symbols. The net result, the sequence of vowel symbols is a property rich in knowledge about the nature of the speech segment.

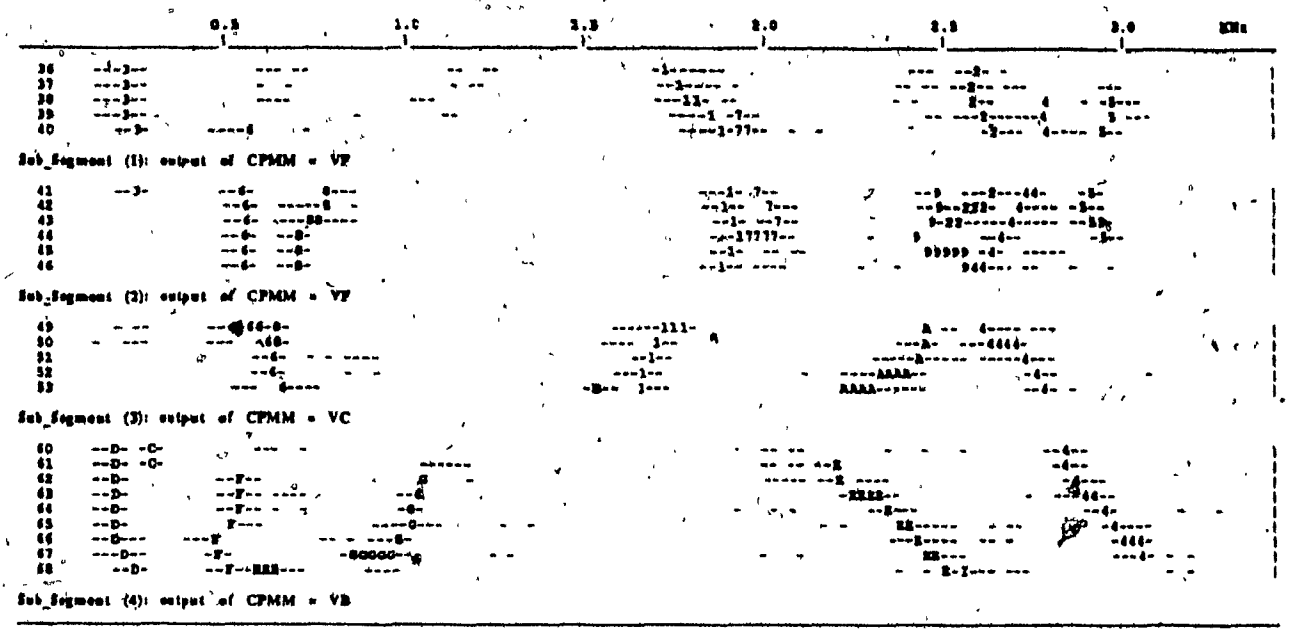


Fig. 7.1b Example of a segment based labeling for digit "zero"

The conclusion is that, frequency domain based CPMMs are proven to be a right tool for describing (ie. labeling) speech segments. The approach is speaker-independent, computationally fast, easy to model, manageable in size and take less effort and time to learn.

7.7 The Scoring Technique

Two important problems to be considered about scoring are: (1) possibility of ruling out the right candidate and (2) the elaborate searching needed for locating the right candidate. Network based models like Harpy, is an example of the latter. Even though some form of heuristics can be used for pruning, such as the beam search used by Harpy, the size of the network is proportional to the number of words in the vocabulary and searching may still be a problem. In such models, pruning is applied while searching, where as, in the proposed Procedural Network model, pruning is applied at every possible stage. "Elimination of totally improbable candidates while keeping all possible candidates" is the philosophy maintained by the main PN as well as by all the sub-networks.

In the SPA sub-network, pre-conditions obtained from the string of vocalic symbols is an example of the above stated philosophy. The pre-condition symbol set consists of all phone candidates which are probable under the given sequence while others are completely eliminated. Solution to the problems mentioned above are obtained this way.

The scoring method itself, the Dynamic Weight Adjustment(DWA) technique, is simple, but, non-conventional. The DWA technique is used to (a) maximize the score of the most probable candidate while considering all possible candidates and (b) obtain a simple, fast, and flexible technique, in which new knowledge towards scoring can easily be added or deleted. Another possible solution for scoring is clearly pure statistical approach which will be discussed in section 7.11c.

The static weights associated initially with each speech unit provide a uniform weight for all phoneme candidates with some expected properties. Later on, based on the results of certain low-level operators, the weights are dynamically adjusted to give the most likely candidate the maximum weight. In some cases no low-level operators are needed and in such situations, scoring will be purely based on the detected properties alone.

Low-level operators are used to obtain detailed speech knowledge when it becomes necessary (i.e., when there are competing candidates evident among the confusion set). These operators are user defined and they can access any component from any level of the hierarchy. Such operators can be simple rules or they can be complicated task verifiers implemented as procedures.

Any number of low-level operators can be used, modified, deleted, or inserted as needed. Based on the results of these operators, weights associated with a particular candidate can be modified easily.

If low-level operators are applied during the early stages of property extraction, then the context in which the knowledge is applied will not be appropriate. For example, consider the operator T_R, for detecting tail ratio. If this operator is applied before classification stage, several candidates from the 3 set would respond, such as,

{o,eu,ai,uai} having T_R = FALSE and,

{eh,ah,ua} having T_R = TRUE.

However, if T_R is applied in the context in which the class number in question

is 4, the confusion will be only among the set {ai,eh,ah} in which, T_R is TRUE for {eh,ah} and FALSE for {ai}. This shows the power of low-level operators if applied in the right context. Also, by applying more such operators, the final hypotheses becomes more robust.

Even though the DWA technique is non-traditional and less mathematical in nature, it is simple to use, easy to implement, and above all, provides good results. Also, this technique leaves open room for any future knowledge upgrading as well as inclusion of more phone-candidates if needed.

7.8 Letters and Digits as Test Data

The objective of this work is to solve one of the key problems in ASR, namely, speaker-independency. However, as an initial step, this task must be accomplished by keeping the problem domain manageable. Recognition of letter-digit vocabulary is a very difficult task, but the domain is manageable. The PN system is capable of handling multisyllable speech, although, the fact that the letter-digit vocabulary set has only few multisyllable words in it, somewhat simplifies the task. However, the letter-digit corpus forms a suitable vocabulary for studying variations in speech sound as well as the solving of the problem of similar types.

This letter-digit vocabulary set contains several words which are identical and also it contains two major subgroups, the plosives, and nasals, both of which are considered to be very difficult to recognize even as a speaker-dependent task. Also, the vocabulary set contains all the phonemes in English as well as examples of allophonic variations and many low level phonological effects. For example, the letter "u" can be pronounced giving the speech unit sequence <VB> or it can be pronounced as eu giving the sequence <VF VB>. Such cases are frequent in the vocabulary set if they are pronounced in an isolated manner.

The letter-digit vocabulary set also illustrates coarticulation effects. A typical example is the digit "zero". This digit, depending on the pronunciation, could produce a monosyllabic or bisyllabic word. The recognition system must be capable of handling such situations.

To conclude, recognition of the letter-digit vocabulary set is a very difficult problem, although the set itself is manageable and all inter- and intra-speaker variations of different sound classes are obtainable.

7.9 Contributions of this Thesis Work

This work is mostly focused on application of vision techniques for capturing speaker independent properties for the recognition of vowels, diphthongs, and vowel like sounds in any given context. The concept is illustrated through a working model. At each level of conception, several novel ideas were presented. The Procedural Network based recognition model also demonstrates the importance of variable depth analysis and integration of knowledge obtained at different levels and in different contexts, for making word hypothesis. Some of the notable contributions are:

a) Treating Speech Spectrogram as Patterns

It has been demonstrated that the Biological Vision approach with perceptual grouping and morphological descriptions are well suited for giving a machine a similar capability as humans of interpreting Speech Spectrograms. Described properties, which are knowledge based, can be learned statistically.

Properties extracted in this way are robust as well as speaker-independent. Such properties are similar to "islands of reliability".

b) The CPMM

In conjunction with the above mentioned approach, a Continuous Parameter and Frequency based Markov Model is used to learn acoustic properties in the quasi-stationary regions of the speech pattern.

Since the CPMM is modeled over the frequency domain, the chain contains small number of states and transitions. Using spectral lines and their relationship (distance) in frequency and energy seems to be the best approach to capture variabilities that are present within quasi-stationary regions. Because of speaker variabilities, spectral lines within these stationary regions may be misplaced or they may exhibit large energy differences for the same speaker, same speech, etc. Variations of this nature are well learned using CPMMs.

c) Internal Segmentation

Large acoustic segments were sub-segmented based upon the stationary behaviour of the lines in the pattern. Acoustic segments were cut into smaller segments

whenever the anchor line become unstable. Stable regions are then labeled using CPMMs. For each AS a sequence of vocalic labels are generated, one for each internal segment. Concatenation of these labels reflects the transitional behaviour of the spoken sound. Knowledge about all possible sequences of symbols belonging each sound can also be learned statistically and later on use this measurement for hypothesis generation.

d) Dynamic Weight Adjustment

Instead of using statistical methods to learn all possible variations for each of the sound classes, a new technique based upon DWA is proposed.

In the Dynamic Weight Adjustment approach, the property vector may contain properties of any type, detected from any context. Any property can be deleted or any new properties can be inserted without the need for any relearning. In addition, the property vector with the static weights can reside outside the software, allowing the user to modify the weights independently.

e) Speaker Independency

In order to achieve speaker independency, the Procedural Network as well as each task performing sub-network in the recognition system must achieve speaker-independency.

The system has been conceived in such a way that each sub-system can be:

- tested for speaker independency,
- new knowledge can be incorporated whenever available,
- free to choose any available technique for extraction and representation of new knowledge.

The SPA sub-system is tested for many speakers and for many context. Context means that the system must recognize vowels or diphthongs which may appear anywhere in the speech sound. Even if the identified candidate is not the top winner, the minimum requirement is that, the right candidate should be among the top 5. In this way, evidence from other sub-systems may lead to the correct recognition. This objective is achieved for the SPA sub-system. Another important point is that, even if the system fails, the reason for failure is usually clear, and new knowledge (tuning process) can easily be incorporated.

To summarize, speech spectrogram interpretation using the vision approach shows promising results. This approach is also quite ideal for the AI based recognition models.

7.10 Future Work

Problems in Artificial Intelligence, unlike problems solved using a conventional algorithmic approach, attempt to give the machine a "sort of intelligence" similar to that used by human to solve problems. Our belief regarding the human information processing system is that it has a long term memory called the cognitive system which uses symbols and symbolic relationship for information processing.

In this work, the SPA sub-system attempts to incorporate a similar approach for speech recognition. Therefore, the future work in this area could involve:

a) *Extending the vision approach to other problems in speech recognition.* The proposed technique could be extended to other types of speech signal which are quasi-stationary in nature, like, nasals and glides. Spectrograms of nasals, liquids, and glides also are quasi-stationary. Fig. 7.2 and 7.3 shows the pattern of nasal sound in letter "m" and "n". The SPA, as it stands now, groups all nasals into one class (eh: {l,m,n}) and it is the task of the nasal-sub-network to solve confusion among "l", "m", and "n". Recognition of elements of the nasal group is a difficult problem, and it may be worth looking into certain perceptual properties for these letters in respective patterns. Again, it may be necessary to transform the pattern into other forms in order to obtain pertinent properties.

b) *Obtaining more symbolic knowledge from the pattern using Pattern Transformation techniques.* For example, we could transform the pattern into its negative form, ie., by considering the "valley" instead of peaks in the spectrum. Fig 7.4a shows the positive and 7.4b shows the negative of Fig. 7.4a. This would allow one to observe the pattern from a different perspective. It is certain that when human observe patterns, the vision sub-system tries to capture information contained by observing the pattern from different perspectives.

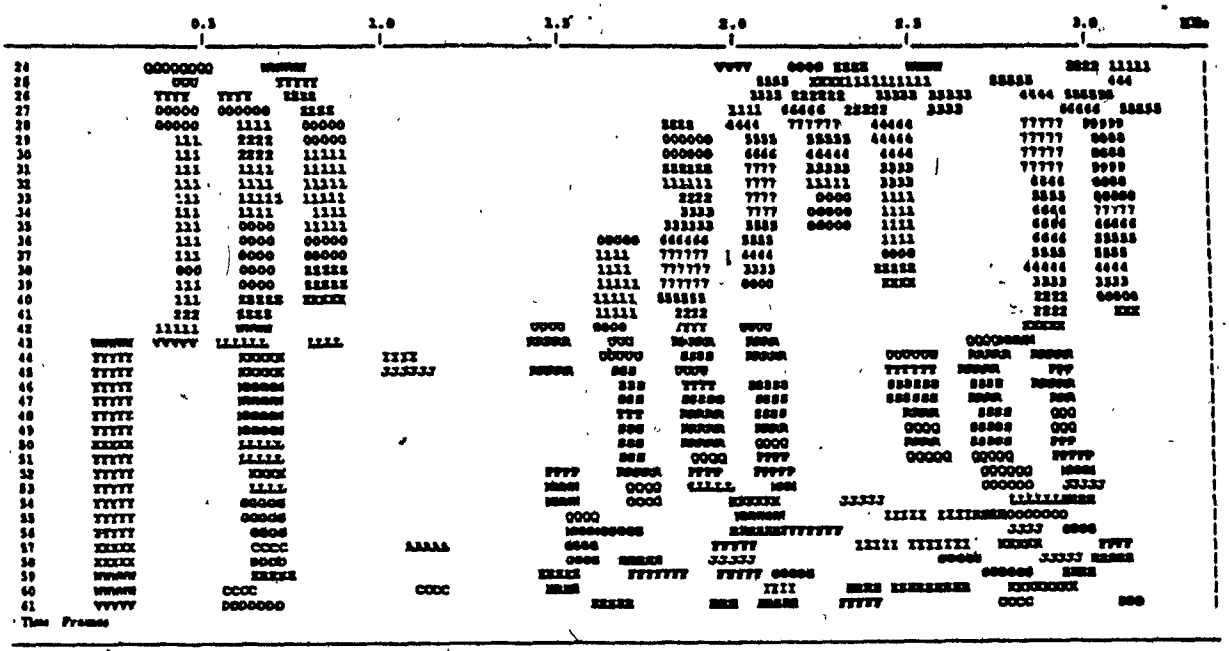


Fig. 7.2 Pattern of a nasal sound in letter "m"

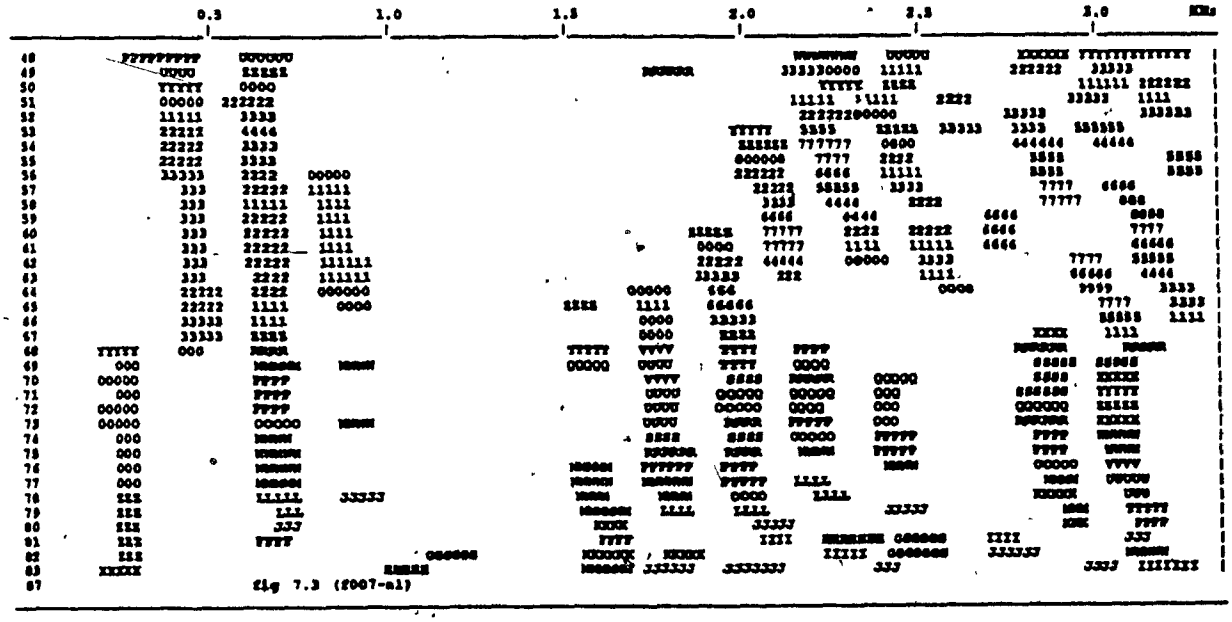


Fig. 7.3 Pattern of a nasal sound in letter "n"

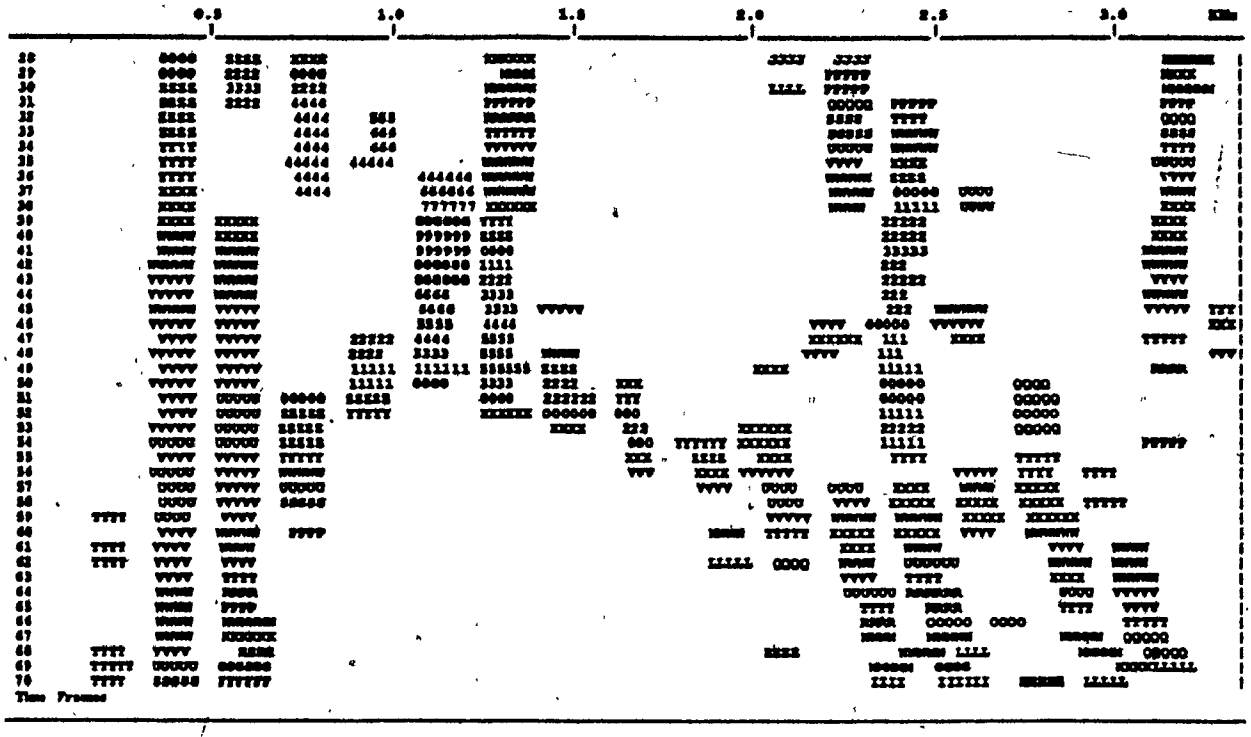


Fig. 7.4a Pattern of the letter "y"

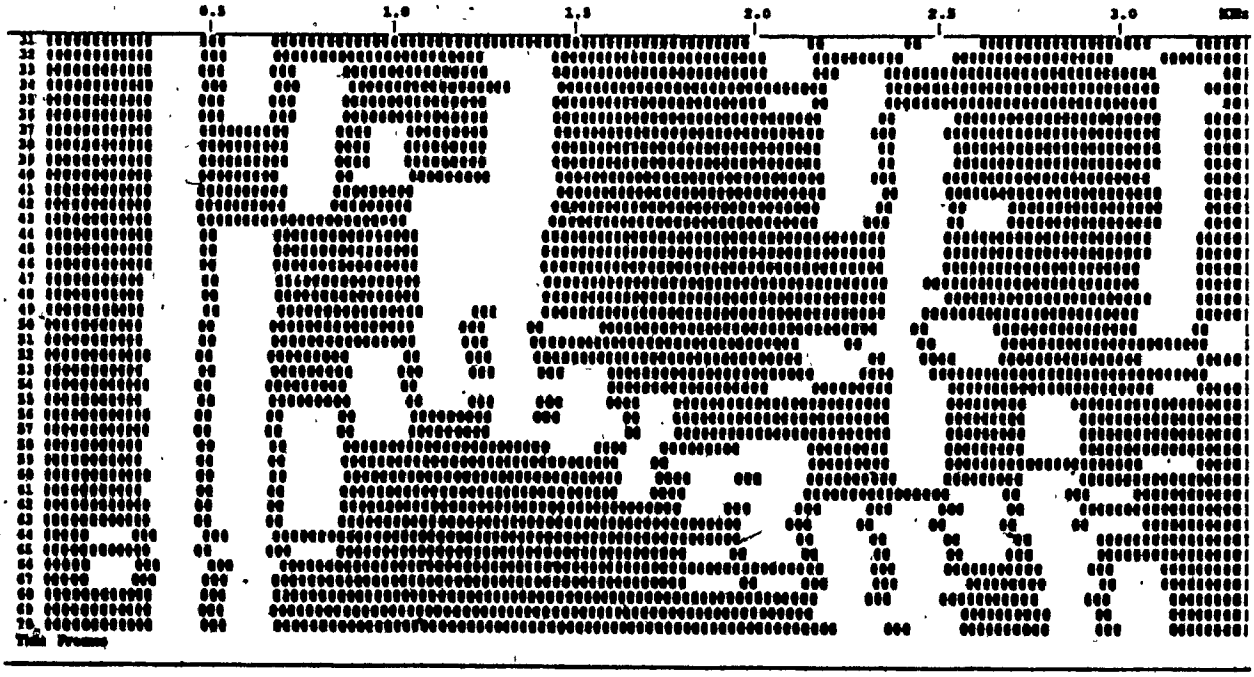


Fig. 7.4b The "negative" of the pattern of Fig. 7.4a

Similarly in Fig 7.4b the dark regions can be described in a symbolic way. Descriptions of this nature may be give more evidence to a given situation, which is worth looking into.

c) *Considering a statistical approach for score evaluation* instead of the proposed DWA approach. Even though it is not yet tested, it is worth looking into a statistically oriented technique for scoring candidates. In order to do this, a large amount of data belonging to each speech unit in the set, {a,e,ai,...}, has to be collected and morphological properties, as described in chapter 3 and chapter 4, have to be learned statistically. Later on during recognition, probabilities can be computed for the incoming property vector.

The performance of SPA is not expected to be better, however, for compatibility reasons, SPA could return scores which are of similar type as those returned by other sub-systems in the Procedural Network system.

In summary, this thesis work has explored the potential of using vision-approaches in speech spectrogram interpretation. The interpreted information on vocalic regions was shown to be a viable approach to speech recognition which is clearly speaker-independent. This work also opens the door for solving similar problems using various techniques mentioned in this thesis.

REFERENCES

[1] W. A. Lea, "The value of Speech Recognition Systems," in *Trends in Speech Recognition*, W. A. Lea, Ed, Englewood Cliffs, NJ, 1979.

[2] R. M. Schwartz, "Acoustic Phonetic Recognition," *6th International Conference of Pattern Recognition*, Munich, pp. 925-965, 1982.

[3] A. V. Oppenheim and R. W. Schafer, "Homomorphic Analysis of Speech," *IEEE Trans. on Audio and Electroacoustics*, Vol. AU-16, No. 2, pp. 221-226, 1968.

[4] R. Jakobson, C. G. M. Fant, and M. Halle, *Preliminaries to Speech Analysis: The Distinctive Features and their Correlates*, Cambridge, Mass.: The MIT Press, 1952.

[5] R-M. S. Heffner, *General Phonetics*, Madison: The University of Wisconsin Press, 1950.

[6] G. Fant, "A note on Vocal Tract size factors and Non-Uniform F-pattern sealings," *Quaterly Progress and status Report 4/66*, Speech Transmission Laboratory, Royal Institute of Technology, Stockholm, pp. 22-30, 1966.

[7] V. W. Zue and R. M. Schwartz, "Acoustic Processing and Phonetic Analysis," Chapter 5, In *Trends in Speech Recognition*, W. A. Lea, Ed, Englewood Cliffs, NJ, 1979.

[8] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, Prentice Hall, Inc., Englewood Cliffs, 1978.

[9] T. Skinner, "Speaker invariant characterizations of Vowels, Liquids, and Glides Using Relative Formant Frequencies," *Journal of Acoustic Society of America*, Vol. 62, supplement 1, pp. 821, 1977.

[10] S. E. Levinson, "Structural methods in Automatic Speech Recognition," *Proceedings of the IEEE*, Vol. 73, No. 11, pp. 1625-1650, 1985.

[11] G. A. Miller, G. A. Heise, and W. Lichten, "The intelligibility of speech as a function of the context of the test materials," *Journal of the Experimental Psycology*, Vol. 41, pp. 329-335, 1951.

[12] P. S. Cohen and R. L. Mercer, "The phonological component of an automatic speech recognition system," in *Speech Recognition*, D. R. Reddy, Ed. New York: Academic Press, 1975, pp. 275-320.

[13] R. Nakatsu and M. Kohda, "An acoustic processor in a conversational speech recognition system", *Rev. ECL*, Vol. 26, pp. 1505-1520, 1978.

[14] W. A. Woods, "Motivation and overview of SPEECHLIS: An experimental prototype for speech understanding research," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-23, pp. 2-10, 1975.

[15] D. W. Shipman and V. W. Zue, "Properties of large lexicons: Implications for advanced isolated word recognition systems," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, Paris, France, pp. 546-549, 1982.

[16] F. R. Chen, "Acoustic-Phonetic constraints in Continuous Speech Recognition: a case study using the digit vocabulary, *Ph. D. thesis*, MIT, 1980.

[17] C. Scagliola, "Continuous speech recognition without segmentation: Two ways of using diphones as basic speech units," *Speech Commun.*, Vol. 2, pp. 199-201, 1983.

[18] A. E. Rosenberg, L. R. Rabiner, J. G. Wilpon, and D. Kahn, "Demisyllable based isolated word recognition system," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-31, pp. 713-726, 1983.

[19] G. Ruske and T. Schotola, "The efficiency of demisyllable segmentation in the recognition of spoken words," in *Automatic Speech Analysis and Recognition: Proc. NATO Advanced Study Institute*, J. P. Haton, Ed. Dordrecht, The Netherlands: Reidel, 1982, pp. 153-163.

[20] R. De Mori, P. Laface, and Y. Mong, "Parallel algorithms for syllable recognition in continuous speech," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-7, pp. 56-69, 1985.

[21] W. A. Lea, M. F. Medress, and T. E. Skinner, "A prosodically guided speech understanding strategy," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-23, pp. 30-38, 1975.

[22] G. Mercier, A. Nouhen, P. Quinton, and J. Siroux, "The KEAL Speech Understanding System," in *Spoken Language Generation and Understanding: Proc, NATO Advanced Study Institute*, Bonas, France, J. C. Simon, Ed. Dordrecht, The Netherlands: D. Reidel, 1979, pp. 525-544.

[23] G. Perennou, "The ARIAL II speech recognition system", in *Automatic Speech Analysis and Recognition: Proc, NATO Advanced Study Institute*, J-P. Haton, Ed., Dordrecht, The Netherlands: D. Reidel, pp. 269-275, 1982.

[24] C. S. Myers and S. E. Levinson, "Speaker independent connected word recognition using a syntax directed dynamic programming procedure," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-30, pp. 561-565, 1982.

[25] D. E. Walker, "The SRI Speech Understanding System," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-23, pp. 397-416, 1975.

[26] H. Bourlard, J. Wellekens, and H. Ney, "Connected digit recognition using vector quantization," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, San Diego, CA, pp. 26.10.1-26.10.4, 1984.

[27] J- P. Haton and J. M. Pierrel, "Syntactic-semantic interpretation of sentences in the MYRTILLE-II speech understanding system," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, Denver, CO, pp. 892-895, 1980.

[28] B. Aldefeld, S. E. Levinson, and T. G. Szymanski, "A minimum distance search technique and its application to automatic directory assistance," *Bell system Tech. Journal*, Vol. 59, pp. 1343-1356, 1980.

[29] L. R. Bahl, J. K. Baker, P. S. Cohen, A. G. Cole, F. Jelinek, B. L. Lewis, and R. L. Mercer, "Automatic recognition of continuously spoken sentences from a finite state grammar," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Washington, DC, pp. 418-421, 1979.

[30] S. E. Levinson, "The effects of syntactic analysis on word recognition accuracy," *Bell Syst. Tech. J.*, Vol. 57, pp. 1627-1644, 1977.

[31] J. K. Baker, "The DRAGON system-An overview," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-23, pp. 24-29, 1975.

[32] F. Jelinek and R. L. Mercer, "Interpolated estimation of Markov source parameters from sparse data," in *Pattern Recognition in Practice*, E. S. Gelsema and L. N. Kanal, Eds. Amsterdam, The Netherlands: North Holland, pp. 381-402, 1980.

[33] L. R. Rabiner, S. E. Levinson, and M. M. Sondhi, "On the application of vector quantization and Hidden Markov models to speaker independent isolated word recognition," *Bell Syst. Tech. J.*, Vol. 62, pp. 1075-1105, 1983.

[34] E. Merlo, R. De Mori, M. Palakal, and G. Mercier, " A continuous Parameter and frequency domain based Markov Model," *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, Tokyo, 1986.

[35] R. De Mori, "A descriptive technique for automatic speech recognition," *IEEE Trans. Audio Electroacoust.*, Vol. AU-21, pp. 89-100, 1973.

[36] T. Kohonen, H. Riittinen, M. Jalanko, E. Reuhkala, and S. Haltsonen,"A thousand word recognition system based on learning subspace method and redundant hash addressing," in *Proc. 5th Int. Conf. on Pattern Recognition*, Miami Beach, FL, pp. 158-165, 1980.

[37] R. K. Moore, "Systems for Isolated and Connected Word Recognition," in *New Systems and Architectures for Automatic Speech Recognition and Synthesis*, R. De Mori and C. Y. Suen, Eds., NATO Advanced Study Institute, Springler-Verlag, 1984.

[38] R. De Mori and D. Probst, "Computer Recognition of Speech," Chapter 20, *in Handbook of Pattern Recognition and Image Processing*, Academic Press, 1986.

[39] J-P. Haton, "Knowledge-Based and Expert Systems in Automatic Speech Recognition," in *New Systems and Architectures for Automatic Speech Recognition and Synthesis*, R. De Mori and C. Y. Suen, Eds., NATO Advanced Study Institute, Springler-Verlag, 1984.

[40] G. M. White and P. J. Fong, "k-nearest-neighbour decision rule performance in a Speech Recognition System," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 5, pp. 389, 1975.

[41] H. Sakoe, "Two-Level DP-Matching-A Dynamic Programming-Based Pattern Matching Algorithm for Connected Word Recognition," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-27, pp. 588-595, 1979.

[42] C. S. Myers and L. R. Rabiner. "A Level Building Dynamic Time Warping Algorithm for Connected Word Recognition," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-29, pp. 284-297, 1981.

[43] H. Ney, "The use of a one stage dynamic programming algorithm for connected word recognition," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-32, pp. 263-271, 1984.

[44] J. K. Baker, "Stochastic modeling for automatic speech understanding," In *Speech Recognition*, D. R. Reddy, Ed. New York: Academic Press, pp. 521-542, 1975.

[45] V. R. Lesser, R. D. Fennel, L. D. Erman, and D. R. Reddy. "Organization of the Hearsay II Speech Understanding System," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-23, pp. 11-24, 1975.

[46] L. D. Erman, D. R. Fennel, R. B. Neely, and D. R. Reddy, "The HEARSAY-I speech understanding system: An example of the recognition process," *IEEE Trans. Comput.*, Vol. C-25, pp. 422-431, 1976.

[47] R. M. Gray, "Vector Quantization", *IEEE ASSP Magazine*, Vol. 1, No. 2, pp. 4-29, 1984.

[48] F. Jelinek, L. R. Bahl, and R. L. Mercer. "Design of a Linguistic Statistical Decoder for the Recognition of Continuous Speech," *IEEE Trans. Infor. Theory*, Vol. IT-21, NO. 5, pp. 250-256, 1975.

[49] F. Jelinek, " Continuous Speech Recognition by Statistical Methods," *Proceedings of the IEEE*, Vol. 64, No. 4, pp. 532-556, 1976.

[50] L. R. Bahl, A. Cole, F. Jelinek, R. L. Mercer, A. Nadas, D. Nahamoo, and M. Picheny, "Recognition of isolated word sentences from a 5000 word vocabulary office correspondence task," In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, Boston, MA, pp. 1065-1067, 1983.

[51] A. Averbuch et al., "Experiments with the Tangora 20,000 word Speech Recognizer," *Proc. Int. Conf. on Acoustics, Speech, and Signal processing*, Dallas, 1987.

[52] L. R. Bhal, F. Jelinek, and R. L. Mercer, "A maximum likelihood approach to continuous speech recognition", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-5, pp. 179-190, 1983.

[53] R. W. Schafer and L. R. Rabiner, "Digital representations of Speech Signals," *Proceedings of IEEE*, Vol. 63, pp. 662-667, 1975.

[54] V. R. Lesser et al., "Organization of the Hearsay II Speech Understanding System," *IEEE Transaction on Acoustics, Speech, and Signal Processing*, Vol. 23, pp. 11-23, 1975.

[55] J. Prager et al., "Segmentation Processes in the Visions Systems," *5th International Joint Conference on Artificial Intelligence*, Cambridge.

[56] J-P. Haton, "Present Issues in Continuous Speech Recognition and Understanding," in *Trends in Speech Recognition*, W. A. Lea, Ed, Englewood Cliffs, NJ, 1979.Hillsdale, NJ: Lawrence Erlbum Assoc., pp. 3-50, 1980.

[57] V. W. Zue, "The use of speech knowledge in automatic speech recognition", *IEEE Proceedings*, pp. 1602-1615, November 1985.

[58] D.H. Klatt, "Review of the ARPA Speech Understanding Project", *Journal of Acoustic Society of America*, Vol. 62, pp. 1345-1366, 1977.

[59] K. N. Stevens, "Acoustic correlates of some phonetic catagories", *Journal of Acoustic Society of America*, Vol. 68, pp. 836-842, 1980.

[60] R. De Mori, R. Gubrynowicz, and P. Laface, "Inference of a knowledge source for the recognition of nasals in continuous speech", *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-27, No. 5, pp. 538-549, October 1979.

[61] P. Demichelis, R. De Mori, P. Laface, and M. O'Kane, "Computer recognition of plosive sounds using contextual information", *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-31, pp. 359-377, 1983.

[62] R. M. Haralick Personal Communication.

[63] L. G. Shapiro, R. M. McDonald, S. R. Sternberg, "Shape recognition with mathematical morphology." In *Proc. 8th Inter. Conf. on Pattern Recognition, IEEE C.N. 86 CH 2342-4*, Paris, France, pp. 416-418, 1986.

[64] H. S. Baird, "Applications of multidimensional search to structural feature identification." In *Proc. Nato Advanced Research Workshop on Syntactic and Structural Pattern Recognition*, Sitges, Spain, October 1986.

[65] F. Jelinek, "The development of an experimental discrete diction recognizer", *IEEE Proceedings*, pp. 1616-1624, November 1984.

[66] H. P. Ni, E. A. Feigeinbaum, J. J. Anton, A. J Rockmore, "Signal-to-symbol transformation. HASP/SLAP case study", *The Artificial Intelligence Magazine*, Vol. 3, No. 2, pp. 23-35, 1982.

[67] R. Waldinger, "Achieving several goals simultaneously" in *Machine Intelligence*, E. Elcock and D. Michie eds., Ellis Horwood, pp. 8, 94-136, 1977.

[68] R. De Mori, L. Lam, M. Gilloux, "Learning and plan refinement in a knowledge-based system for automatic speech recognition" *IEEE Trans. Pattern Anal. Machine Intell.*, to appear.

[69] K. S. Fu, Syntactic pattern recognition and applications, *Prentice Hall*, 1982.

[70] M. Palakal et al., "Automatic recognition of spoken vowels and diphthongs in many contexts and for many speakers.", In preparation.

[71] J. Rouat et al., "Automatic recognition of spoken consonants in many contexts and for many speakers.", In preparation.

[72] G. Kopec, and M. Bush, "Network-based isolated digit recognition using vector quantization", *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. ASSP-33, pp. 850-856, August 1985.

[73] K. F. Lee, "Incremental Network Generation in Word Recognition.", *Proceedings IEEE ASSP Int. Conference*, pp. 77-80, Tokyo, Japan, April 1986.

[74] R. A. Cole, R. M. Stern, M. S. Phillips, S. M. Brill, P. Specker, and A. P. Pilant. "Feature-based speaker-independent recognition of English letters.", *Proceedings IEEE ICASSP 83*, pp. 731-734, 1983.

[75] L. D. Erman, F. Hayes-Roth, V. R. Lesser, and D. R. Reddy, "The HEARSAY-II speech understanding system, integrating knowledge to resolve uncertainty," *ACM Computing Surveys*, Vol. 12, pp. 213-253, 1980.

[76] J. N. Larar, "Lexical access using broad acoustic-phonetic classifications," *Computer Speech & Language*, Vol. 1, No. 1., pp. 47-59, 1986.

[77] G. Shichman, et al., "An IBM PC based large-vocabulary isolated-utterance speech recognizer," *Proceedings ASSP IEEE Int. Conference*, pp. 53-56, Tokyo, Japan, April 1986.

[78] D. W. Shipman and V. W. Zue, "Properties of large lexicons: implications for advanced isolated word recognition systems," *Proceedings IEEE ICASSP 82*, pp. 546-549, 1982.

[79] V. W. Zue and R. A. Cole, "Experiments on Spectrogram Reading," *IEEE International Conference on Acoustics, Speech and Signal Processing*, Wahsington D. C., pp. 116-119, 1979.

[80] V. W. Zue and L. F. Lamel, "An Expert Spectrogram Reader: A Knowledge-based Approach to Speech Recognition," *IEEE International Conference on Acoustics, Speech and Signal Processing*, Tokyo, pp. 1197-1200, 1986.

[81] R. A. Cole, A. I. Rudnicky, V. W. Zue, and D. R. Reddy, "Speech as Patterns on Paper," In *Perception and Production of Fluent Speech*, R. A. Cole, Ed., Hillsdale, NJ: Lawrence Erlbum Assoc., pp. 3-50, 1980.

[82] R. A. Cole and V. W. Zue, "Speech as Eyes See It," In *Attention and Performance VIII*, R. S. Nickerson, Ed., Hillsdale, NJ: Lawrence Erlbum Assoc., pp. 475-494, 1980.

[83] J. Johannson, J. MacAllister, T. Michalek, and S. Ross, "A Speech Spectrogram Expert," *IEEE International Conference on Acoustics, Speech and Signal Processing*, Boston, MA, pp. 746-749, 1983.

[84] P. E. Stern, M. Eskenazi, and D. Memmi, "An Expert System for Speech Spectrogram Reading," *IEEE International Conference on Acoustics, Speech and Signal Processing*, Tokyo, pp. 1193-1196, 1986.

[85] N. Carbonnell, D. Fohr, J-P. Haton, and F. Longchamp, "An Expert-System for the Automatic reading of French Spectrogram," *IEEE International Conference on Acoustics, Speech and Signal Processing*, Boston, MA, 1983.

[86] D. G. Lowe, Perceptual Organization and Visual Recognition, *Kluwer Academic Publishers*, Massachusetts, 1985.

[87] A. Triesman, "Perceptual Groupings and Attention in Visual Search for Features and for Objects," *Journal of Experimental Psycology: Human Perception and Performance*, Vol. 8, No. 2, pp. 194-214, 1982.

[88] J. Beck, "Effect of orientation and of shape similarity on perceptual groupings," *Perception and Psychophysics*, Vol. 1, pp. 300-302, 1966.

[89] J. Beck, "Perceptual grouping produced by line figures," *Perception and Psychophysics*, Vol. 2, pp. 491-495, 1967.

[90] D. Katz, Gestalt Psychology: Its Nature and Significance, *New York: Ronald Press Co.*, 1950.

[91] M. Wertheimer, "Untersuchungen zur lehe von der Gestalt II," *psychol. Forschung*, No. 4, 1923. Translated as "Principles of preceptural organization," in *Readings in Perception*, Deardslee and M. Wertheimer, Eds., Princeton, NJ, pp. 115-135, 1958.

[92] A. P. Witkin and J. M. Tenenbaum, " On the role of Structure in Vision," in *Human and Machine Vision*, Beck, Hope, and Rosenfeld, Eds, New York: Academic Press, pp. 481-543, 1983.

[93] N. J. Naccache and R. Shinghal, "SPTA: A proposed algorithm for thinning binary patterns," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-14, No. 3, pp. 409-419, 1984.

[94] S. W. Zucker, "Computational and Psychological Experiments in Grouping: Early Orientation Selection," in *Human and Machine Vision*, Beck, Hope, and Rosenfeld, Eds, New York: Academic Press, pp. 481-543, 1983.

[95] R. O. Duda and P. E. Hart, "use of the Hough transformation to detect lines and curves in pictures," *Communications of ACM*, Vol. 15, No. 1, pp. 11-15, 1975.

[96] L. R. Rabiner and B. H. Juang, "An Introduction to Hidden Markov Models," *IEEE Acoustics, Speech and Signal Processing Magazine*, No. 1, pp. 4-16, 1986.

[97] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An Introduction to the

application of the theory of probabilistic functions of a Markov process to automatic speech recognition," *Bell Syst. Tech. J.*, Vol. 62, pp. 1035-1074, 1983.

[98] G. Kopec, "Formant tracking using Hidden Markov models," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 1113-1116, 1985.

[99] A. Holbrook and G. Fairbanks, "Diphthong Formants and thier Movements," *Journal of Speech and hearing Research*, No. 3, pp. 38-58, 1962.

[100] L. E. Baum, "An inequality and associated maximization technique in the statistical estimation for probabilistic functions of Markov processes.", *Inequalities*, Vol. 3, pp. 1-8, 1972.

Appendix-A The Dyanamic Weight Adjustment Table

Class Number	Code Number	Candidate	Property	Weight
1	101	a	v_f	0.8
1	101	e	v_f	0.8
1	101	ei	fdn	0.6
1	101	eu	dsnd	0.8
2	201	ah	v_c	0.8
2	201	eh	v_c	0.8
2	201	ua	v_c	0.6
2	201	ai	v_c	0.6
2	201	uai	v_c	0.6
2	201	ah	asnd	0.8
2	201	eh	asnd	0.8
2	201	ai	asnd	0.5
2	201	ua	asnd	0.5
2	201	uai	asnd	0.5
2	201	ah	stcr	0.8
2	201	ai	stcr	0.6
2	201	ua	stcr	0.6
2	201	uai	stcr	0.6
2	201	ah	stcl	0.8
2	201	eh	stcl	0.8
2	201	ai	stcl	0.6
2	201	ua	stcl	0.6
2	201	uai	stcl	0.6
3	301	u	v_b	0.6
3	301	o	v_b	0.8
3	301	eu	v_b	0.6
3	301	u	stcl	0.6
3	301	o	stcl	0.6
3	301	u	dsnd	0.6
3	301	o	dsnd	0.6
3	301	u	fdn	0.8
3	301	o	fdn	0.8
4	401	ai	v_f	0.9
4	401	ai	v_c	0.9
4	401	eh	v_f	0.7
4	401	eh	v_c	0.7
4	401	ah	v_f	0.7
4	401	ah	v_c	0.7
4	401	uai	v_f	0.7
4	401	uai	v_c	0.7
4	401	ai	asnd	0.6
4	401	ai	dsnd	0.6
4	401	uai	asnd	0.5
4	401	uai	dsnd	0.5
4	401	eh	asnd	0.6
4	401	ah	asnd	0.6
4	401	eh	stcl	0.6
4	401	ah	stcl	0.6

Appendix-A The Dyanamic Weight Adjustment Table (contd.)

Class Number	Code Number	Candidate	Property	Weight
4	401	ai	stcr	0.8
4	401	ai	stcl	0.8
4	401	uai	stcr	0.6
4	401	uai	stcl	0.6
4	402	ah	v_f	0.9
4	402	ah	v_c	0.9
4	402	eh	v_f	0.9
4	402	eh	v_c	0.9
4	402	ai	v_f	0.6
4	402	ai	v_c	0.6
4	402	uai	v_f	0.6
4	402	uai	v_c	0.6
4	402	ai	asnd	0.6
4	402	ai	dsnd	0.6
4	402	uai	asnd	0.5
4	402	uai	dsnd	0.5
4	402	eh	asnd	0.6
4	402	eh	stcl	0.8
4	402	ah	asnd	0.6
4	402	ah	stcl	0.8
4	402	ai	stcr	0.6
4	402	ai	stcl	0.6
4	402	uai	stcr	0.6
4	402	uai	stcl	0.6
4	403	ai	asnd	0.9
4	403	ai	dsnd	0.8
5	501	eu	v_f	0.9
5	501	eu	v_b	0.9
5	501	eu	fdn	0.8
5	501	eu	stcl	0.6
5	501	eu	dsnd	0.8
5	502	o	v_f	0.9
5	502	o	v_b	0.9
5	502	o	nsph	0.8
5	502	o	stcl	0.6
5	502	o	dsnd	0.8
6	601	uai	asnd	0.8
6	602	uai	asnd	0.6
6	602	ua	asnd	0.6
6	603	uai	v_b	0.6
6	603	ua	v_b	0.8
6	604	uai	v_c	0.8
6	604	ua	v_c	0.6
7	701	eh	v_c	0.9
7	701	eh	v_b	0.9
7	701	o	v_c	0.9
7	701	o	v_b	0.9
8	801	uai	asnd	0.8
8	802	uai	v_b	0.6
8	802	ua	v_b	0.8

Appendix-A The Dynamic Weight Adjustment Table (contd.)

Class Number	Code Number	Candidate	Property	Weight
8	803	uai	v_c	0.8
8	803	ua	v_c	0.6
8	804	uai	v_f	0.6
8	804	ua	v_f	0.8
8	805	uai	v_f	0.8
8	805	ua	v_f	0.6
9	901	xl	v_f	0.9
9	901	xl	v_b	0.9
9	901	xl	v_c	0.9
9	901	xl	stcr	0.6
9	901	xl	stcl	0.6
10	1001	ei	v_f	0.8
10	1001	ei	fdn	0.6
10	1001	ei	v_c	0.8
10	1001	ai	v_c	0.6
10	1001	ai	v_f	0.6
10	1001	ei	asnd	0.6
10	1002	ai	v_f	0.8
10	1002	ai	fdn	0.6
10	1002	ai	v_c	0.8
10	1002	ei	v_c	0.8
10	1002	ei	v_f	0.8
10	1002	ei	asnd	0.6
51	5101	o	v_f	0.4
51	5101	xl	stcr	0.6
51	5101	o	dsnd	0.6
51	5101	o	stcl	0.6
51	5101	eh	v_f	0.4
51	5101	eh	dsnd	0.6
51	5101	eh	stcl	0.6
52	5201	ua	v_b	0.6
53	5301	eh	v_f	0.4
53	5301	a	v_o	0.4
53	5302	a	v_f	0.6
53	5303	uai	asnd	0.6
53	5303	ai	asnd	0.6
53	5304	xl	v_f	0.6
53	5304	ua	v_f	0.6
53	5305	uai	v_f	0.6
53	5305	ai	v_f	0.6
53	5305	eh	v_c	0.6
53	5306	xl	v_c	0.6
53	5306	ua	v_c	0.6
53	5306	eh	v_f	0.6
53	5307	uai	v_c	0.6
53	5307	ai	v_c	0.6
54	5401	eh	v_f	0.4
54	5402	uai	asnd	0.6
54	5402	ai	asnd	0.6
54	5403	xl	v_f	0.6

Appendix-A The Dyanamic Weight Adjustment Table (contd.)

Class Number	Code Number	Candidate	Property	Weight
54	5403	ua	v_f	0.6
54	5404	uai	v_f	0.6
54	5404	ai	v_f	0.6
54	5404	eh	v_b	0.5
54	5405	xl	v_c	0.6
54	5405	ua	v_c	0.6
54	5405	eh	v_c	0.5
54	5406	uai	v_c	0.6
54	5406	ai	v_c	0.6
55	5501	eh	v_f	0.4
55	5501	e	v_b	0.4
55	5502	ua	asnd	0.6
55	5502	eu	asnd	0.2
55	5503	xl	v_f	0.6
55	5503	ua	v_f	0.6
55	5504	eu	v_f	0.6
55	5504	e	v_f	0.6
55	5504	eh	v_b	0.5
55	5505	xl	v_b	0.6
55	5505	ua	v_b	0.6
55	5505	eh	v_f	0.6
55	5506	eu	v_b	0.6
55	5506	e	v_b	0.6
56	5601	eh	v_f	0.4
56	5602	uai	asnd	0.6
56	5602	ai	asnd	0.6
56	5603	uai	v_f	0.2
56	5603	ai	v_f	0.2
57	5701	ah	v_f	0.4
57	5701	ah	v_b	0.4
57	5702	uai	asnd	0.6
57	5702	ai	asnd	0.6
57	5703	uai	v_f	0.2
57	5703	ai	v_f	0.2
58	5801	uai	asnd	0.6
58	5801	ai	asnd	0.6
58	5802	xl	v_b	0.6
58	5802	ua	v_b	0.6
58	5803	uai	v_b	0.6
58	5803	ai	v_b	0.6
58	5804	xl	v_c	0.6
58	5804	ua	v_c	0.6
58	5805	uai	v_c	0.6
58	5805	ai	v_c	0.6
59	5901	ah	v_c	0.8
59	5901	eh	v_c	0.6
59	5902	eh	v_c	0.8
59	5902	ah	v_c	0.6
60	6001	eu	v_f	0.9
60	6001	eu	v_b	0.9

Appendix-A The Dyanamic Weight Adjustment Table (contd.)

Class Number	Code Number	Candidate	Property	Weight
60	6001	eu	fdn	0.8
60	6001	eu	stcl	0.6
60	6001	eu	dsnd	0.8
60	6001	o	v_f	0.9
60	6001	o	v_b	0.9
60	6001	o	nsph	0.8
60	6001	o	stcl	0.6
60	6001	o	dsnd	0.8
61	6101	eh	v_f	0.9
61	6101	eh	v_c	0.9
61	6101	ai	v_f	0.7
61	6101	ai	v_c	0.7
61	6101	ah	v_f	0.7
61	6101	ah	v_c	0.7
61	6101	eh	stcr	0.6
61	6101	ah	stcr	0.6
61	6101	ai	stcr	0.5
61	6102	ai	v_f	0.9
61	6102	ai	v_c	0.9
61	6102	eh	v_f	0.7
61	6102	eh	v_c	0.7
61	6102	ah	v_f	0.7
61	6102	ah	v_c	0.7
61	6102	eh	stcr	0.5
61	6102	ah	stcr	0.5
61	6102	ai	stcr	0.6