COMPARISON OF SEVERAL DISTANCE MEASURES FOR SEGMENTATION AND ISOLATED WORD RECOGNITION

by

Ralph W. Brown

B.S.E.E. North Carolina State University
Summa Cum Laude
(1980)

SUBMITTED TO THE DEPARTMENT OF
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

MASTER OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1982

Signature of Author ..... Signature redacted
Department of Electrical Engineering and Computer Science, April 26, 1982.

Certified by ..... Signature redacted
Victor W. Zue, Thesis Supervisor

Accepted by .....
Chairman, Departmental Committee on Graduate Studies

COMPARISON OF SEVERAL DISTANCE MEASURES FOR SEGMENTATION AND ISOLATED WORD RECOGNITION

by

Ralph W. Brown

Submitted to the Department of Electrical Engineering and Computer Science on April 26, 1982 in partial fulfillment of the requirements for the Degree of Master of Science in Electrical Engineering

ABSTRACT

Dynamic Programming has been demonstrated to be a very effective technique for isolated word recognition. Multi-pass and level building generalizations of Dynamic Programming have also been suggested for connected word recognition. One of the major drawbacks to Dynamic Time Warping is the excessive storage and computational requirements for large vocabularies. If a method of data reduction can be developed to reduce these requirements the application of this powerful technique to larger and more difficult recognition tasks may become feasible. This thesis explores a technique of data reduction which makes use of segmentation based on acoustic similarity. A generalized segmentation system is discussed and a specific implementation of this system is used in performing the segmentation. Segments in this system are represented by a single parameter vector, in this case the average over all the parameter vectors contained in the segment. Three methods of dealing with the segmented representations are proposed and are expressed within the mathematical formalism of the normal DTW algorithm. A comparison is made of the relative performance of these methods, one which uses Dynamic Time Warping directly on the segmented speech ignoring durational information, another which incorporates segment duration by weighting local distances in the Dynamic Time Warping algorithm by the average segment duration, and one which expands segments by effectively repeating the average frame to the number of frames contained in the original speech segment, to determine what trade offs exist between data reduction, computational efficiency, and recognition accuracy.

To demonstrate the advantages of this segmentation and data reduction technique three distance measures, Itakura's log likelihood ratio, cosh, and cepstral, are evaluated to determine their relative performance isolated word recognition. The one giving the highest recognition accuracy is used in experiments using segmentation. The experimental work is conducted using a 30 word calculator task vocabulary. Six repetitions of this vocabulary from each of four speakers, two male and two female, are used in a speaker dependent system. The results of these experiments show that the cosh distance measure gives the highest recognition accuracy in the unsegmented case. A data reduction of almost 50% and a savings in computation time of 52% can be obtained without significantly reducing recognition accuracy using the method of expanding segments. This method gives significantly better recognition performance than the other two methods for all levels of data reduction, but is poorer in terms of computation. The trade offs between data reduction, recognition accuracy, and computation are discussed in the thesis.

THESIS ADVISOR: Dr. Victor W. Zue
TITLE: Assistant Professor of Electrical Engineering

ACKNOWLEDGEMENTS

I would like to express my gratitude to Dr. Victor W. Zue for his guidance, assistance, and patience in advising me on this thesis. Despite some of the rougher moments of this endeavor, I do feel that I have benifited greatly from this experience and my association with Dr. Zue. His influence has caused me to rethink many of my goals and ambitions, setting them higher than I had previously considered.

I would especially like to thank my family and friends, without whose loving support and encouragement none of this would have been possible. My greatest indebtedness without question is here.

Additional thanks should be given to the members of the MIT Speech Group who provided a pleasant and enjoyable environment in which to undertake my research.

To my parents, George and Mary Brown.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

TABLE OF SYMBOLS

$a_j^{(i)}$	j^{th} linear prediction coefficient of i^{th} iteration in Durbin's recursion
a_j	j^{th} linear prediction coefficient
$\bar{a}_j$	j^{th} linear prediction coefficient of the segmental average
$A(z)$	optimum predictor for a frame of speech data
α	minimum residual error for test frame
b_j	j^{th} autocorrelation coefficient of the predictor polynomial
$\bar{b}_j$	j^{th} autocorrelation coefficient of the predictor polynomial for the segmental average
c_j	j^{th} cepstral coefficient
$c_j(m)$	j^{th} cepstral coefficient of the m^{th} frame
$\bar{c}_j$	j^{th} cepstral coefficient of the segmental average
$\mathbf{C}(m)$	comparison frame for the m^{th} frame
$d(\mathbf{x},\mathbf{y})$	distance between parameter vectors **x** and **y**
$d(\mathbf{R}(n),\mathbf{T}(m))$	distance between frame n of the reference and frame m of the test
$\tilde{d}(n,m)$	local distance between the n^{th} element of the reference and the m^{th} element of the test
D	minimum normalized total distance along any path
$D_A(n,m)$	accumulated distance to the point (n,m) along the best path
δ	residual error for a frame of data through inverse filter for a different frame of data
$E^{(i)}$	i^{th} order linear prediction residual
E_{MAX}	maximum slope constraint on a path
E_{MIN}	minimum slope constraint on a path
F_m	cut off frequency for anti-aliasing filter
F_s	sampling frequency
g_n	intermediate variable for deriving cepstral coefficients
$G(L^R(n),L^T(m))$	weighting function based on segment lengths

$H(z)$	transfer function of a system
$(i(k), j(k))$	parameterized path functions
k_j	j^{th} PARCOR coefficient
$\bar{k}_j$	j^{th} PARCOR coefficient of the segmental average
L	frame rate, in samples
L	number of coefficients used in cepstral distance measure
L_p	the set of distance metrics defined by the integral of the spectral error raised to the power p
$L(k)$	length in number of frames of k^{th} segment
$L_T(k), L_R(k)$	segment length, in frames, for test and reference templates
m_i	intial frame of a segment
m_f	final frame of a segment
M	length of a test pattern, in frames
N	length of a reference pattern, in frames
$\mathbf{N}(\tilde{W})$	normalization factor for weighting function $\tilde{W}$
p	order of an LPC production
R	range limit, in frames
$R(i)$	i^{th} autocorrelation lag
$R(m,i)$	i^{th} autocorrelation lag for the m^{th} frame
$\bar{R}(i)$	i^{th} autocorrelation lag for the segmental average
$\mathbf{R}(m)$	m^{th} frame of a reference pattern
$\tilde{\mathbf{R}}(k)$	k^{th} segment of a segmented reference pattern
$s(n)$	speech signal
$\tilde{s}(n)$	pre-emphasized speech signal
$S_{MAP}(m)$	segmental mapping function, maps frame number to segment number
$S^T_{MAP}(m), S^R_{MAP}(m)$	segmental mapping function for test and reference templates
σ	gain of the LPC spectral model
T_s	sampling period

$\mathrm{T}(m)$	m^{th} frame of a test pattern
$\tilde{\mathrm{T}}(k)$	k^{th} segment of a segmented test pattern
$u(\mathrm{L})$	cepstral distance measure truncated to L coefficients
$V(\theta)$	spectral difference on log magnitude scale
$w(n)$	window function
$\tilde{W}(k)$	weighting function for local distances
$x(n)$	windowed speech signal

1. INTRODUCTION

1.1 Automatic Speech Recognition

With the ever growing presence of computers in society today, there has been considerable interest in improving the quality of person-machine interaction. By improving the quality of this relationship broader and more efficient use of these tools can be expected. Verbal communication is one of the most natural means of conveying ideas and information among people. Studies have indicated higher communication rates for speech than written or typed means[1]. Since it is the most common method of communication for humans and does not require special training, such as typing or key-punching, it would seem that speech is the most natural means for people to communicate with machines. Speech communication would also make interaction with computers more human and perhaps less intimidating than other means. Automatic Speech Recognition (ASR) by computers is thus a major goal in improving the quality of person-machine interaction. Many applications of ASR have been suggested, from automatic control of assembly line machinery to aids for the handicapped[2] [3].

There has been a great deal of research in the area of ASR, much of this having been initiated by the Advanced Research Projects Agency with the ARPA-SUR project. Research in this field has shown that there are several different classes of problems which all fall under the heading of Automatic Speech Recognition. Reddy[4], in an overview, outlines some of the factors involved in determining the different classes of speech recognition and discusses much of the research that has been carried out in these areas. One of the major factors which classifies speech recognition problems is whether the input is connected speech or words spoken one at a time. The problems which are incurred by connected speech make this class of problems significantly more difficult than for isolated words and phrases. Other factors which classify speech recognition problems include, vocabulary size, questions of speaker dependency, recording environment, and possible restrictions on grammar and syntax.

Isolated word recognition differs from connected speech recognition in that it restricts the

speech input to be words from a predefined vocabulary spoken with distinct pauses between them, hence the name isolated word recognition, or sometimes discrete utterance recognition. While this is a very unnatural way of speaking, it does remove many of the coarticulatory effects that occur when words are spoken in a continuous fashion and greatly simplifies the problem of recognition. In general the vocabulary size will be relatively small, on the order of 10 to 100 words. Martin[2] found that speaking rates between 30 and 70 words per minute on the average with peak rates close to 120 words per minute can be achieved for isolated words or phrases. Many possible applications of such limited speech recognition systems have been suggested.

Connected speech recognition problems remove the constraint of inserting pauses between words and allow connected or continuous speech as input. Often the desired vocabulary for these systems will be considerably larger than for an isolated word recognition system. These two factors make these problems considerably more complex than for isolated word recognition. It is beyond the scope of this thesis to attempt to solve the more difficult problems of connected speech recognition and thus the work undertaken in this thesis falls under the category of isolated word recognition. However, many of the techniques discussed in this work have application in other areas of speech processing.

1.2 Isolated Word Recognition as Pattern Matching

Isolated Word Recognition has traditionally been approached as a pattern matching problem where the input is compared to a dictionary of stored references to determine the closest match. The block diagram of an isolated word recognition system is shown in Figure 1.1. Typically the reference dictionary is built up through a training process in which one or more repetitions of each word in the vocabulary are recorded for use as the reference templates (patterns). Normally these patterns are time varying sequences of parameter vectors, each vector representing a windowed section, or frame, of the speech signal. Often the templates for several different speakers will be combined in some way to form reference patterns for a speaker independent system[5]. Once a reference dictionary has been established recognition is accomplished by comparing the unknown

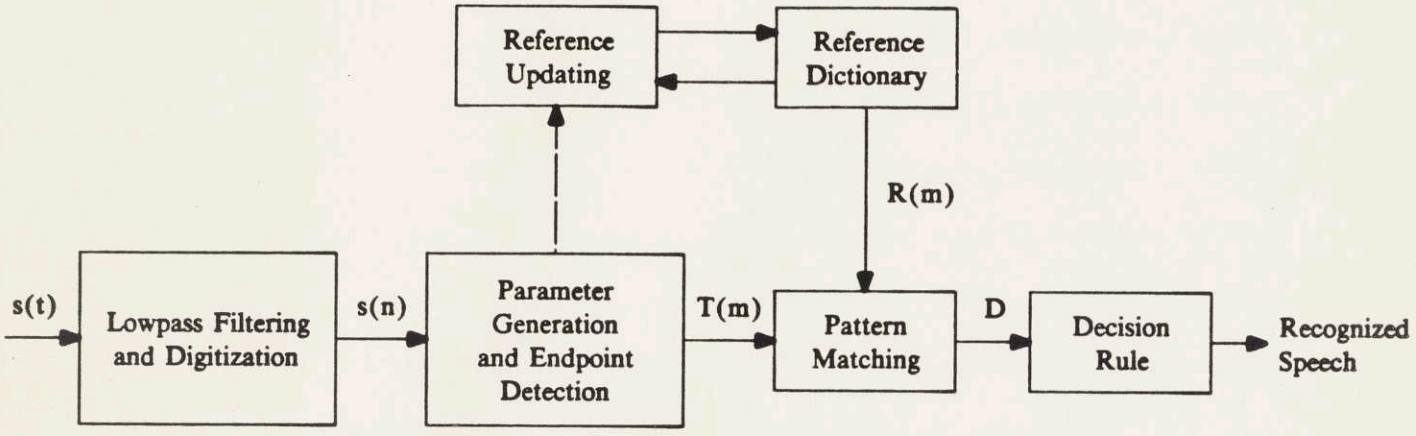


Figure 1.1. Isolated Word Recognition System (after Myers et.al.)

input, or test, utterance to each template in the dictionary and the closest match is given as the recognized utterance.

One of the most critical factors involved comparing two utterances is their time alignment. Since words or phrases are rarely spoken twice with exactly the same duration some form of time alignment must be used before a measure of similarity between two utterances can be obtained. Several approaches to the time alignment problem have been applied for isolated word recognition, one of the most promising of these is a Dynamic Programming technique known as Dynamic Time Warping (DTW). The application of Dynamic Programming to speech recognition was initially proposed by Sakoe and Chiba[6] in 1971. Since that time many researchers have demonstrated that this technique performs extremely well[7] [8] [9] [10], in some cases achieving recognition accuracies of better than 99%[9][10]. This method achieves its high recognition accuracy by allowing a non-linear time alignment between two utterances, compensating for local time variations in the speaking rate. The basic principle behind Dynamic Time Warping is to find the optimum time alignment between the test and reference templates using a minimum total distance criterion.

One of the major drawbacks to Dynamic Time Warping is the amount of storage required for the reference dictionary. It is necessary to store at least one reference template for each word in the vocabulary. As a result the amount of storage required for larger vocabularies can be excessive. Dynamic Time Warping is also computationally intensive, since the time alignment process must be carried out against each of the reference templates. In general the amount of storage and computation required will increase linearly with vocabulary size. These two requirements can place severe constraints on the size of the vocabulary. A method of reducing these requirements must be developed to enable the application of this technique to larger and more difficult speech recognition problems.

Several techniques have been suggested for reducing computation in the DTW algorithm, such as absolute range limiting[9] or choosing a locally optimum path [11]. While these techniques can substantially reduce the amount of computation, they do not reduce the amount of storage required

for the reference templates. Tappert and Das[12] suggest a technique for reducing both storage and computation requirements by creating the speech patterns in a manner that is not unlike variable frame rate vocoding. In this technique only those frames which differ significantly from the last stored frame are used in the speech template. Their work shows that savings of as much 50% to 60% in storage and a reduction in computation by a factor of 4 to 6 can be achieved without significantly reducing recognition accuracy.

The results given by Tappert and Das indicate that much of the redundant information contained within the steady state regions of an utterance are not of vital importance to the recognition process. By eliminating this redundant information both storage and computation can be significantly reduced. In order to implement this approach it is necessary to identify the steady state regions, i.e. those regions that show little change in the spectral characteristics, and represent them by a reduced amount of data. In other words the utterance must be segmented into transitional regions and regions of steady state.

1.3 Segmentation for Data Reduction

There are basically two forms of segmentation that might be considered, phonetic and acoustic. Phonetic segmentation attempts to segment an utterance into phonetic units, such as phonemes, syllables, or demisyllables. Acoustic segmentation isolates units of acoustic homogeneity. Many attempts have been made in the area of phonetic segmentation[13] [14] [15]. Unfortunately none have performed flawlessly and often several alternative possible segmentations of an utterance must be proposed[16]. It is generally accepted[17] that phonetic segmentation is not possible on the basis of acoustic information alone and often a detailed knowledge of the relations between phonology, articulation, and acoustics must be applied to form a more reasonable segmentation. This higher level phonetic knowledge is often implemented as phonetic or syntactic rules, which are usually heuristic in nature and can be difficult to implement. Often the performance of such knowledge sources can be unreliable. Our understanding of phonology and speech production is far from complete and further research must be carried out in these areas before reliable phonological rules

can be accurately stated.

Within the context of data reduction the need for accurate phonetic segmentation is less of a concern. For example, when the main objective of the segmentation process is to accurately delineate phonetic units it is not acceptable to have extremely short segments. However, when the objective is to accurately represent significant changes in the acoustic signal, segments consisting of only a few frames of data are perfectly acceptable. Since the major concern in data reduction is to preserve the acoustically significant information, i.e. the transitional information, one is primarily interested in capturing the frames where there is a significant change in the acoustical characteristics. Most phonetic units contain both steady state and transitional information giving rise to a wide range of variation in the acoustical characteristics within the units themselves. For example, stop consonants will usually have a brief period of silence, the stop gap, before release of the stop burst. Combining these two acoustically diverse regions as one segment obscures the more important transitional information. The variation of acoustical characteristics is even greater across a syllable. Phonetic segmentation in general will segment regions containing significant variation in acoustical characteristics and thus acoustic segmentation is better suited for our approach to data reduction.

Once the utterance has been segmented into units of acoustic homogeneity there is no need to retain all of the data contained within a segment since much of it is redundant information. Data reduction can then be achieved by storing each of these segments as a single representative vector. This representative vector might be computed as the average of the parameter vectors for all the frames of speech contained within the segment. Using a single representative vector is a convenient method of representing a speech segment since a segment composed of several frames is handled in the same way as an isolated frame. The same processing techniques which are applied to unsegmented speech templates can be applied to segmented speech templates formed in this manner.

1.4 Scope of Thesis Research

It is the objective of this thesis to demonstrate the advantages of data reduction by acoustic segmentation and determine what tradeoffs exist between storage, computation, and recognition accuracy. One of the most significant factors affecting recognition accuracy in this approach is how the information contained in the segment durations is incorporated into the Dynamic Time Warping algorithm. Three methods of handling durational information are evaluated and the results for each are discussed. Another, lesser objective of this thesis is to determine the relative performance of three distance measures proposed by Gray and Markel[18]. These distance measures are evaluated for their performance prior to acoustic segmentation and the one yielding the highest recognition accuracy is used in experiments involving segmentation.

The remainder of this thesis is organized in the following way. In chapter two the basic principles of acoustic segmentation are discussed and a generalized segmentation system is proposed. The basic design choices in the segmentation system are presented and the choices used in the actual implementation are given. In chapter three the principles of Dynamic Time Warping are presented and the effects of segmentation on Dynamic Time Warping are discussed. Three alternative methods for making use of durational information are proposed in the last section of chapter three. In chapter four the speech parameters and distance measures used in these experiments are discussed. Chapter five contains explanations of the experiments carried out and their results. Chapter six contains conclusions and recommendations for future work.

2. ACOUSTIC SEGMENTATION

Acoustic segmentation is based on the assumption that transitions between phonetic sounds in an utterance will exhibit large changes in the spectral or parametric representations of the speech waveform and within phonetic sounds there will be relatively little spectral change. Thus if a measure of acoustic similarity, or distance, is applied to adjacent or neighboring frames of speech data the resulting contour will show peaks in the areas of sharp spectral change and valleys in the areas of little of no spectral change. Figures 2.1 and 2.2 show plots of distance and energy versus time for the words "B" and "recall". One can, for example, hypothesize segment boundaries at those points in the distance contour where a predetermined threshold is exceeded or at peaks in the distance contour. Goldberg, et.al.[19] describe a parameter independent segmentation system which uses this principle. The major advantage of this approach is its simplicity. It only requires two basic components, a parametric representation of the speech waveform and a metric or distance measure over this parameter space. While this method of segmentation does not perform well in generating phonetic segments without the use of some form of phonetic rules, it is well suited for segmentation into units of acoustic similarity.

2.1 A Generalized Acoustic Segmentation System

To better understand the underlying principles of acoustic segmentation a general model of the algorithm should be discussed. Consider the segmentation system shown in block form in Figure 2.3. In this system the digitized speech signal, $s(n)$, is processed by digital techniques to generate a parameter vector, $\mathbf{R}(m)$, every L samples. A measure of similarity is computed between the current frame, $\mathbf{R}(m)$, and the comparison frame, $\mathbf{C}(m)$, using a distance measure over the parameter space. A boundary decision is made on the basis of the distance $d(\mathbf{R}(m),\mathbf{C}(m))$, and the appropriate action is taken by the segmental representation scheme to generate the segmented representation, $\tilde{\mathbf{R}}(k)$. A segmental mapping function, $S_{MAP}(m)$, is created which is a monotonic nondecreasing function relating the frame number, m, to the segment number, k. Figure 2.4 gives an example of such a mapping function.

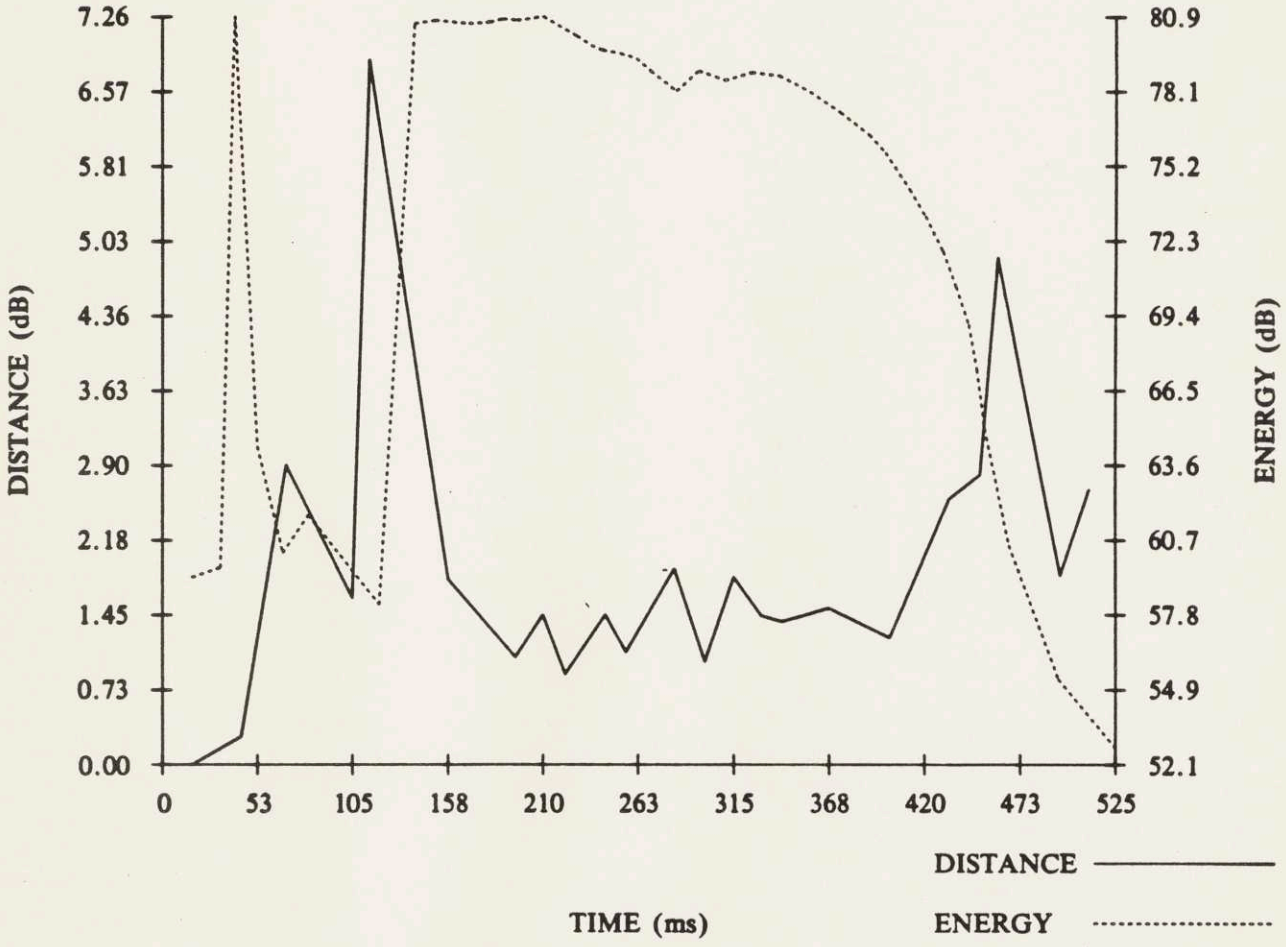


Figure 2.1. Distance and Energy Contours for the Word 'B'

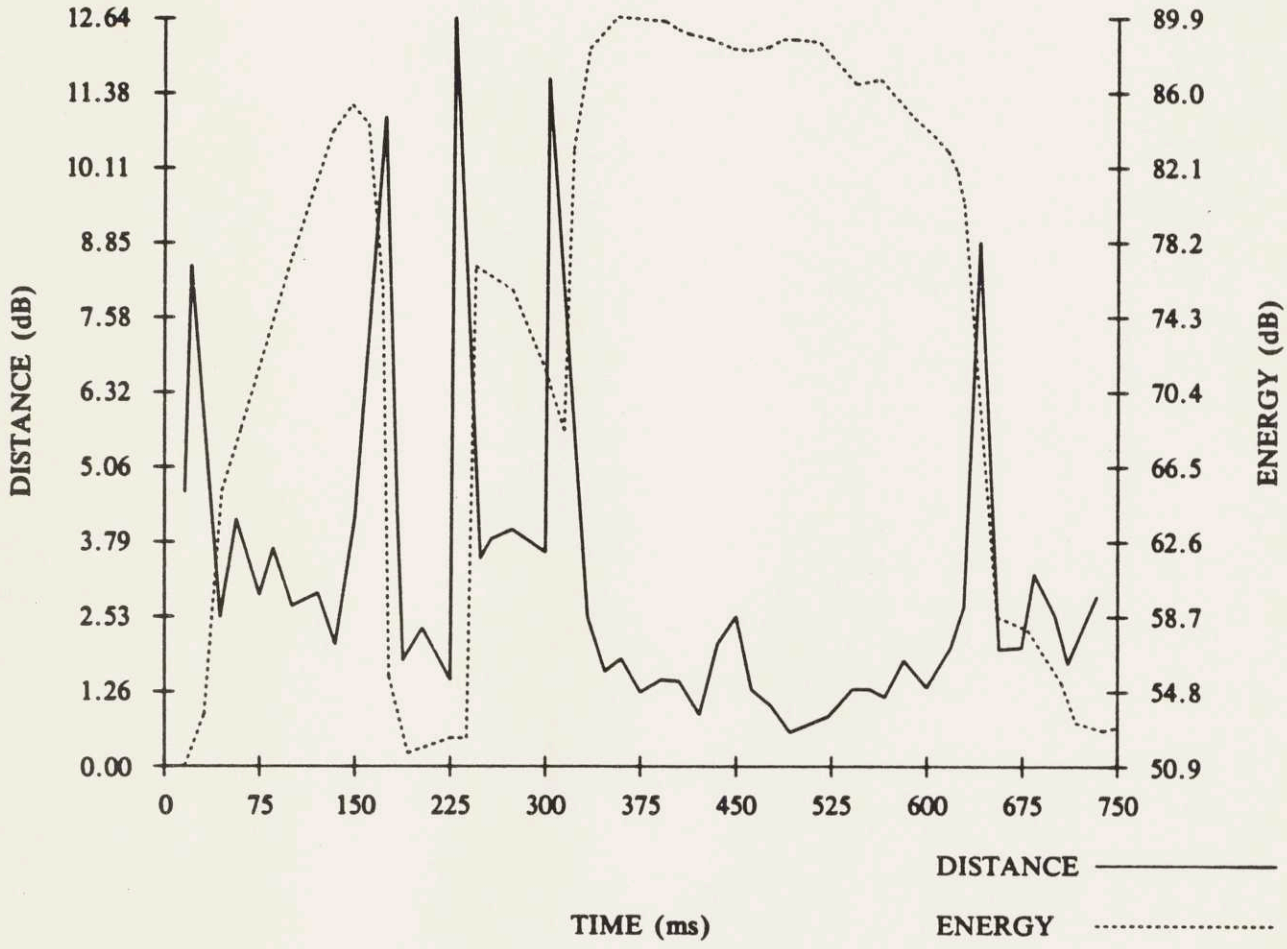


Figure 2.2. Distance and Energy Contours for the Word 'recall'

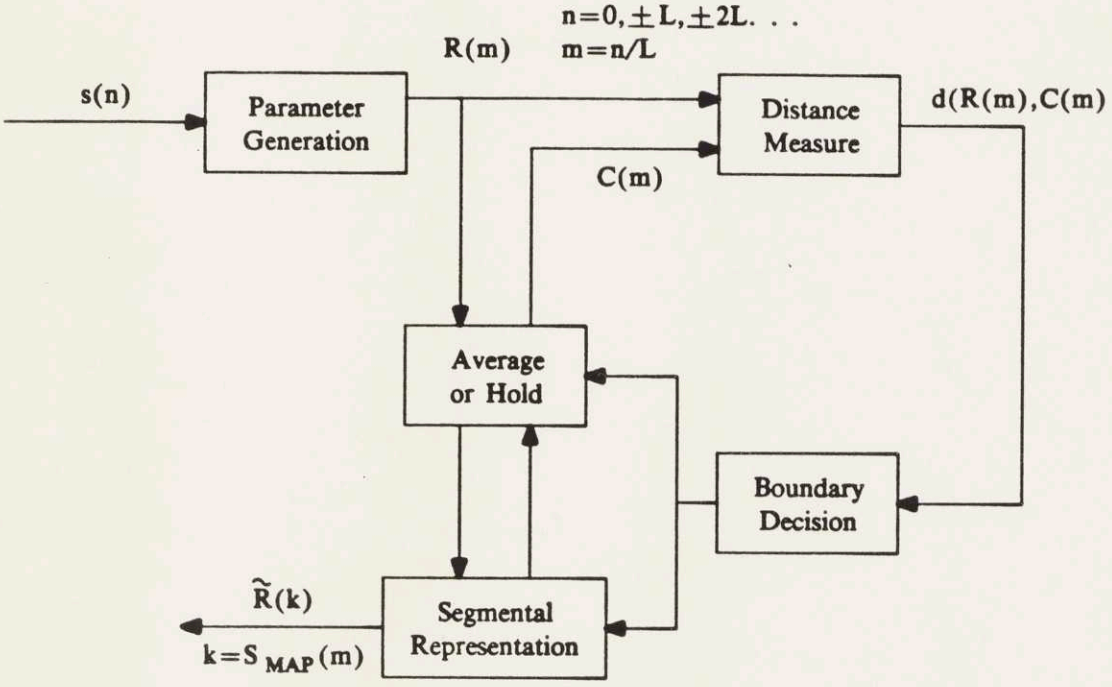


Figure 2.3. Generalized Segmentation System

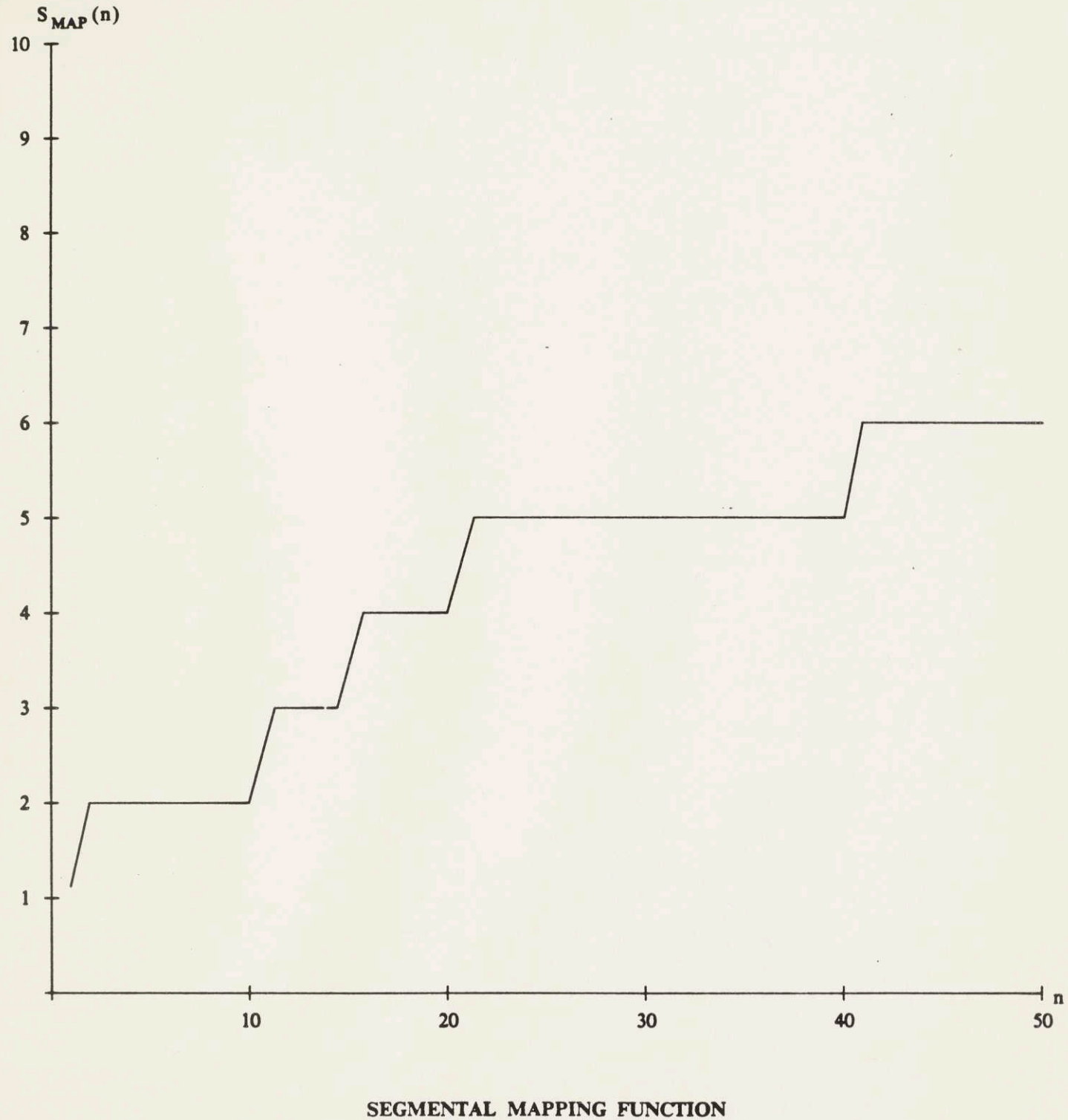


Figure 2.4. Example of Segmental Mapping Function

This model indicates several of the areas in which major design decisions which must be made in order to implement a segmentation system of this form. The major implementation details include the following areas:

1. Parameter Generation, $\mathbf{R}(m)$,
2. Frame Rate, L,
3. Distance Measure, $d(\mathbf{x},\mathbf{y})$,
4. Comparison Frame, $\mathbf{C}(m)$,
5. Boundary Decision,
6. Segmental Representation.

Design choices in each of these areas will affect the performance of the segmentation system. The effects of these factors, possible design choices for each, and the choices made in the actual implementation are discussed in the following sections.

2.1.1 Parameter Generation

Factors involved in parameter generation include, the choice of the parameter set, the frequency at which the speech signal is sampled, whether the speech is pre-emphasized or not, and the size and shape of the analysis window. Unquestionably the most significant of these is the choice of the parameter set. There are many different parameter sets for speech processing that have been discussed in the literature and there is considerable debate as to which may be the best in terms of accurately representing the important aspects of the speech signal. Zue and Schwartz[20] list a few of the more popular speech parameters used in speech processing. Rabiner and Schafer[21] present a very thorough discussion of methods for computing many of these speech parameters. It is important that the speech parameters adequately reflect the characteristics of the speech signal that are of interest. An often used and quite reasonable approach is to require that the parameters give an accurate spectral representation of the signal. Some parameter sets which fit this description include DFT representations, critical-band spectral representations,

homomorphically smoothed spectrum, or cepstrum, and linear prediction spectral estimate. Other speech parameters include zero-crossing rate and overall signal energy. The technique of linear prediction is often used because it can be efficiently computed and provides a reasonably accurate spectral estimation which can be specified by a relatively small number of parameters. A cepstral representation is frequently used as well, since only a few-low order coefficients are necessary to provide a good spectral representation. Because of their wide acceptance and frequent use, linear predictive coding, LPC, and a cepstral representation derived from this LPC representation were chosen as our parameter sets. These choices were also motivated by the choice of distance measures we wished to investigate. A minimal set of parameters for the LPC representation are the inverse filter coefficients, a_k, and the autocorrelation coefficients, $R(k)$. However, additional components are normally added to reduce the amount of computation for the LPC distance measures. These are the minimum residual energy, α, and the autocorrelation coefficients of the inverse filter polynomial, $b(i)$. The cepstral parameter set is usually the $p+1$ low order coefficients, c_k, where p is the order of the LPC inverse filter from which they are derived. Computation of these parameters and some of their properties will be discussed in more detail in chapter four.

Choices for the other factors involved in parameter generation are primarily dictated by the theory of digital signal processing. For example, the sampling frequency, F_s, is given by the well known sampling theorem, which states that the speech signal must be bandlimited to a maximum frequency, F_m, and the sampling frequency must be at least twice this maximum frequency, i.e.

$$F_s \geq 2 \cdot F_m . \tag{2.1}$$

Although the bandwidth of the telephone communication system is roughly 3 kHz and it provides quite intelligible speech, there is a considerable amount of information, especially for female speakers, contained in the speech signal above 3 kHz. It was decided a slightly wider bandwidth should be used in our system. The speech signal was then lowpass filtered to $F_m = 4.8$ kHz and sampled at $F_s = 10$ kHz.

The effect of pre-emphasizing the speech signal is to cancel the combined effect of the glottal wave shape and the radiation characteristics and to compensate for the spectral tilt that is associated with them. The speech signal is pre-emphasized by passing the digitized signal through a first order system with a transfer function of the form

$$H(z) = 1-\mu z^{-1} \tag{2.2}$$

where μ is the pre-emphasis factor. The pre-emphasized speech is thus related to the original signal by

$$\tilde{s}(n) = s(n)-\mu s(n-1) \tag{2.3}$$

Typically the pre-emphasis factor will be in the range $0.9 \leqslant \mu \leqslant 1$. Rabiner et.al.[22] evaluated the effects of varying several of the analysis parameters in an LPC based isolated word recognition system. In comparing the recognition performance for no pre-emphasis, $\mu = 0$, and pre-emphasis with $\mu = 0.95$ they concluded that the use of pre-emphasis was preferred. In our case a pre-emphasis factor of $\mu = 0.95$ was used.

The effects of window shape and size have been discussed in detail by Rabiner and Schafer[21]. In general a rectangular window is considered unacceptable due to its poor frequency characteristics. A tapered window is usually preferred as it weights the central part of the frame of speech data, the portion that is to be represented most accurately, more than the edges and it eliminates discontinuities at window edges. The choice of window length is a trade off between resolution in the time and frequency domains. In short time Fourier analysis good temporal resolution results from a short window while good frequency resolution requires a long window. In linear prediction longer windows increase the computation required for computing the autocorrelation coefficients, however, the window must include several pitch periods to insure an accurate spectral estimate. Typical window lengths used in LPC analysis using the autocorrelation method vary from 10 to 40 ms. The work of Rabiner et.al.[22] demonstrates that the length of the analysis window does not significantly affect recognition performance in an LPC based isolated word recognition system. In our system a 256 point (25.6 ms) Hamming window was used, i.e.,

$$w(n) = 0.54 - 0.46\cos\left[\frac{2\pi n}{N-1}\right], \quad 0 \leqslant n \leqslant N-1 \tag{2.4}$$

where $N=256$.

2.1.2 Frame Rate

The frame rate L is the number of samples that the analysis window is advanced in computing the speech parameters. The frame rate can be expressed in terms of absolute time when it is divided by the sampling frequency, i.e. L/F_s. The frame rate is an important factor in speech processing because it determines the temporal resolution of the speech parameters. Since speech is a non-stationary signal the speech parameters vary over time; the ability to capture rapid changes in these parameters is of critical importance. The lower the frame rate is the better the temporal resolution that can be obtained. However, the amount of data required to represent the utterance increases correspondingly. One would like to choose the lowest possible frame rate which still gives adequate resolution of the speech parameters. Frame rates that have been discussed in the literature typically vary between 10 and 20 ms. Rabiner, et.al.[22] discuss the effects of varying the frame rate within the context of isolated word recognition. In their evaluation, a frame rate of 15 ms yielded the highest recognition accuracy. While it is not clear that this measure of performance is appropriate to segmentation, it does indicate that this frame rate gives an adequate temporal resolution of the speech parameters. For this reason a frame rate of $L=150$ (15 ms) was chosen for our system.

2.1.3 Distance Measures

The distance measure is a critical factor in the segmentation process due to the fact that the boundary decision is based upon the distance $d(\mathbf{R}(m),\mathbf{C}(m))$. Consequently the distance measure should be sensative to changes in the parametric representation and reflect the magnitude of these changes accordingly. Much research has been done in the area of distance metrics for speech processing[18][23] [24] [25]. Some of this effort has been made in developing perceptually consistent distance measures, which give high correlation to results of experiments in perceived acoustical similarity in humans[24][25]. The choice of a distance measure will in part be determined by the

parameter set.

Our interest focused on the three distance measures presented by Gray and Markel[18] and a comparison of the relative performance of them. All of these distance measures are based on LPC derived parameters and are referred to as the Itakura distance measure (or log likelihood ratio), the cosh distance measure, and the cepstral distance measure. Their work discusses the derivation of these distance measures and relates them to the L_2 norm, also known as the rms log spectral difference. A more thorough discussion of these distance measures will be taken up in a later section.

2.1.4 Comparison Frame

The choice of comparison frame is important to the segmentation process because the segment boundary decision is based on the distance between this frame and the current frame, $\mathbf{R}(m)$. It should be noted that more than one comparison frame could be included in the distance used for the boundary decision, as was the case in the segmentation system proposed by Goldberg et.al.[19]. In their work the distance for the previous frame and the frame two analysis intervals prior were included in the boundary decision, so as to capture both rapid and more gradual changes. The additional computation costs in using more than one comparison frame were considered to outweigh any potential gain from doing so. In addition, it was not clear how the relative importance of the distances from each of several comparison frames should be weighed. Thus in our system only one comparison vector is used in computing the distance for the boundary decision.

Given this decision there are still several possible choices for the comparison frame, but only a few logical ones for the purposes of acoustic segmentation. One would expect better performance from the segmentation algorithm if the comparison frame is representative of the current segment. That is, if the current frame should be considered the beginning of a new segment it should differ significantly from the frames which make up the current segment. There are three choices which seem to be reasonable given this restriction. The comparison frame could be an adjacent or neighboring frame, the first frame of the segment, or the average of all the frames contained in the seg-

ment so far. These choices can be written as:

1. neighboring frame,

$$\mathbf{C}(m) = \mathbf{R}(m-j), \quad j \geq 1 \tag{2.5}$$

2. first frame of segment,

$$\mathbf{C}(m) = \mathbf{R}(m_i) \tag{2.6}$$

3. average frame,

$$\begin{aligned} \mathbf{C}(m) &= AVERAGE\,[\mathbf{R}(m_i), \mathbf{R}(m_i+1), \ldots, \mathbf{R}(m)] \\ &= \bar{\mathbf{R}}(m_i, m) \end{aligned} \tag{2.7}$$

where m_i is the initial frame of the segment. The averaging function is not written as the arithmetic mean, because the average vector for some parameter sets, in particular the LPC parameter set, cannot be obtained from the arithmetic average. Computation of the average vector for these parameter sets will be covered in detail in chapter four. Each of these choices will give rise to slightly different behavior in the distance contour $d(\mathbf{R}(m), \mathbf{C}(m))$ and consequently in the segmentation.

There is a certain amount of simplicity in using a neighboring frame. The average or hold operation is a first-in-first-out buffer that operates independent of the boundary decision. By using the adjacent or a neighboring frame more importance is given to local changes in the parametric representation. The choice for j determines, to a certain extent, what rate of change in the parametric representation that is to be weighed most in the segmentation. The value of j is restricted to be positive to insure that the segmentation system is causal and can be implemented directly. Larger values of j will capture more gradual changes, such as what might occur during a transition from a sonorant to a vowel. However, the greater j the greater the storage required to retain past frames. Choosing the first frame of the segment as the comparison frame will limit the absolute change that may occur from the segment initial frame. This will capture the more gradual changes, but there is some question as to whether the segment initial frame is most representative of the current segment. The final choice is attractive because it is similar to a hypothesis test

where the measure is how far the current frame is from the mean of the segment so far. This choice unfortunately has significant overhead due to the fact that the average frame must be computed at the frame rate. The algorithm which gives the simplest implementation and at the lowest computational cost, which is the one used in this study, is to use the previous frame as the comparison vector, i.e. $\mathbf{C}(m) = \mathbf{R}(m-1)$. We shall see in the following section how this choice also provides a simple means of controlling the amount of segmentation that takes place, given the proper choice for the boundary decision algorithm.

2.1.5 Boundary Decision

It is the task of the boundary decision algorithm to make a binary decision at each frame instant as to whether the current frame should begin a new segment or not. In many speech recognition systems boundary decisions will be based on heuristic methods which consider many different factors and parameters of the speech signal. For example, the segmentation and labeling system discussed by Weinstien et.al.[14] considers formant trajectories and dips in the energy contour in making boundary decisions. However, many of these methods are difficult to implement and are more typically used in phonetic segmentation systems. It was our desire to develop a simple algorithmic approach to acoustic segmentation and the use of heuristic methods was discarded in favor of more direct means. As was suggested earlier the important factor in acoustic segmentation is to isolate segments of acoustic similarity. For this reason the boundary decision is based on the distance $d(\mathbf{R}(m),\mathbf{C}(m))$ alone. By considering the characteristic behavior of the distance contour two relatively simple boundary decision algorithms can be suggested. As was discussed in section 2.1, we can expect peaks in the distance contour at those points where there is a significant change in the spectral characteristics. A peak picking or a thresholding algorithm would seem to be an adequate method of determining segment boundaries. Each of these methods has its own advantages and disadvantages.

A peak picking algorithm indicates a segment boundary at local maxima of the distance contour. Since not all such maxima are expected to be significant a more discriminating peak picking

algorithm, such as Mermelstein's[26] *convex hull* algorithm, could be used. Goldberg et.al.[19] used a peak picking algorithm which considered several additional factors, such as the slope of the distance contour and the area beneath the peaks. The advantage of a peak picking algorithm is that it can be made to detect gradual and less drastic changes in the spectral characteristics. This is often desirable since transitions between different phonetic sounds will give rise to peaks in the distance contour of different amplitudes. The disadvantages of a peak picking algorithm is that it is more difficult to implement than a simple thresholding algorithm and it is also difficult to control the amount of segmentation produced.

A thresholding algorithm indicates a boundary whenever $d(\mathbf{R}(m),\mathbf{C}(m))$ exceeds a predetermined threshold. By adjusting this threshold the amount of segmentation, and the resulting data compression, can be easily controlled. Using a comparison frame which does not introduce feedback into the boundary decision, i.e. one which is independent of the boundary decision, the necessary thresholds for different compression ratios can be determined from the statistics of a set of training data. For example, if an adjacent or neighboring frame is used as the comparison frame a histogram of the distance contour can be generated from a set of training data which accurately reflects the characteristics of the distance $d(\mathbf{R}(m),\mathbf{C}(m))$. The necessary threshold to achieve a desired compression can be determined from this histogram. Since a major objective of this thesis was to demonstrate how recognition accuracy varies with percent compression, the thresholding algorithm was chosen over the peak picking algorithm.

2.1.6 Segmental Representation

As was suggested in section 1.3 the desired goal of data reduction can be achieved by careful selection of the segmental representation. Since acoustic segmentation isolates segments which show very little variation in the acoustic characteristics through out, a single parameter vector should adequately represent an entire segment. Representing segments in this form can substantially reduce the amount of data storage and computation required for isolated word recognition. There is little motivation for changing the speech parameters in creating the segmented representa-

tion since these parameters are already available and are assumed to provide a good spectral representation. Using a single representative parameter vector from the original parameter set also provides a convenient means of comparing recognition performance for differing amounts of segmentation and for various DTW algorithms. Creating speech templates in this fashion allows one to apply the same processing techniques to both the segmented and unsegmented templates.

With this concept in mind consideration must be given to the choice for the representative vector. The more accurately the representative vector reflects the characteristics of segment the better the recognition accuracy that could be expected. As with the comparison frame there are several reasonable choices for the representative vector. These fall into two categories, a single selected frame in the segment, or an average of the parameter vectors contained within the segment. Reasonable choices for the first category are the first frame of the segment, the last frame, or the middle frame. These can be written as:

1. first frame,

$$\tilde{\mathbf{R}}(k) = \mathbf{R}(m_i) \tag{2.8}$$

2. last frame,

$$\tilde{\mathbf{R}}(k) = \mathbf{R}(m_f) \tag{2.9}$$

3. middle frame,

$$\tilde{\mathbf{R}}(k) = \mathbf{R}(m_i + (m_f - m_i)/2) \tag{2.10}$$

It is not clear that any of these choices is superior. However, one could argue that the middle frame more clearly represents the steady state characteristics of the phoneme than the first or last frames. There is always the risk in this approach that the selected vector is not the most representative for the segment. Consequently all of these choices were discarded in favor of using an averaged vector, i.e.

$$\begin{aligned} \tilde{\mathbf{R}}(k) &= AVERAGE\,[\mathbf{R}(m_i), \mathbf{R}(m_i+1), \ldots, \mathbf{R}(m_f)] \\ &= \bar{\mathbf{R}}(m_i, m_f) \end{aligned} \tag{2.11}$$

It was felt that the average of all the frames in the segment was more representative of the overall

characteristics of the segment.

There is an additional piece of information that is of interest in processing segmented data, that is the length of the segments. Without this additional piece of information it is impossible to determine what the actual length of the original utterance was. We shall see later how this durational information can be used in interpreting the segmented data. To retain this information an additional component, $L(k)$, containing the number of frames that are represented by the segment was added to the parameter vector. The representative vector now becomes

$$\tilde{\mathbf{R}}(k) = \begin{bmatrix} L(k) \\ \bar{\mathbf{R}}(m_i, m_f) \end{bmatrix}. \tag{2.12}$$

Note that, in our implementation, as the threshold is lowered the amount of segmentation and resulting data compression are reduced. If the threshold is set to zero then the representation reverts to the unsegmented case and the segmental mapping function becomes an identity relationship, i.e.

$$S_{MAP}(m) = m \tag{2.13}$$

and the length of all segments becomes one,

$$L(k) = 1, \quad k=1,2,\ldots,K. \tag{2.14}$$

For convenience this representation was also used for the unsegmented templates and percent compression is written as

$$compression = \frac{\#\ of\ frames - \#\ of\ segments}{\#\ of\ frames} \cdot 100\% \tag{2.15}$$

$$= \frac{N - S_{MAP}(N)}{N} \cdot 100\%$$

where N is the number of frames contained within the utterance.

2.2 Summary

We have discussed a generalized acoustic segmentation system and presented the design decisions made for the implementation used in our experiments. Many of the decisions that were made

in the actual implementation of the segmentation system were made on the basis of simplicity and ease of implementation. The resulting implementation allows a simple means to accurately control the amount of data reduction, which will be used to compare recognition performance. Table 2.1 shows the design choices made in the actual implementation of the segmentation algorithm. In the following chapter we will discuss the principles of Dynamic Time Warping and how the Dynamic Time Warping algorithm can be modified for the segmented speech templates.

Design Choice	Implementation	
	LPC	LPC derived Cepstrum
Parameter Set	α, a_i, $R(i)$, $b(i)$, $1 \leq i \leq p$	c_i, $0 \leq i \leq L$
Sampling Frequency	F_s = 10 kHz	
Pre-emphasis Factor	μ = 0.95	
Window Function	256 point (25.6ms) Hamming window	
Frame Rate	L = 150 samples (15ms)	
Comparison Frame	$\mathbf{C}(m) = \mathbf{R}(m-1)$	
Distance Measures	log likelihood ratio, cosh, cepstral	
Boundary Decision	thresholding algorithm	
Representative Vector	$\bar{\mathbf{R}}(k) = AVERAGE[\mathbf{R}(m_i),\ldots,\mathbf{R}(m_f)]$	

TABLE 2.1. Design Choices for Segmentation System

3. DYNAMIC TIME WARPING

The block diagram of an isolated word recognition system is shown in Figure 3.1. The most important task in this process is the pattern matching algorithm, which in general constitutes the time alignment of the test and reference utterances. The time variation from utterance to utterance, and even between tokens of the same utterance, is such that some time normalization must take place before a measure of similarity between them can be made. Two techniques have traditionally been used to accomplish this time alignment, linear compression or expansion, and a Dynamic Programming technique referred to as Dynamic Time Warping (DTW). Linear compression or expansion will stretch or shrink the time scales of the utterances linearly so that they are of the same duration. Dynamic Time Warping allows the amount of compression along the time axes to vary in a nonlinear fashion to achieve the time alignment. While linear time alignment will align the endpoints of the utterances perfectly, the internal features of an utterance will often be misaligned causing errors in recognition. Dynamic Time Warping uses Dynamic Programming to achieve an optimal time alignment by allowing greater or less compression in those areas where it is required, as illustrated Figure 3.2. White and Neely[10] demonstrate that the DTW technique performs substantially better than linear time alignment even when several alternative endpoints were used in the linear matching. In general DTW has been shown to give exceptional recognition performance for many vocabularies [7][8][9][10].

The purpose of Dynamic Time Warping is to determine the optimum time alignment between the unknown, or test pattern, $\mathbf{T}(m)$, $m=1,2,\ldots,M$, and a reference pattern, $\mathbf{R}(n)$, $n=1,2,\ldots,N$. These patterns are in general multidimensional time varying parameter vectors, as might be generated by the segmentation algorithm. These time sequences of parameter vectors are discrete in nature, i.e. the speech parameters are computed at distinct time intervals and represent a given segment of speech data. The criterion that is used in Dynamic Time Warping for determining the "optimum" time alignment between test and reference patterns, or templates, is the minimum distance path along a surface defined by the distances between test frames and reference frames. Again, as in acoustic segmentation, the distance is some measure of acoustic

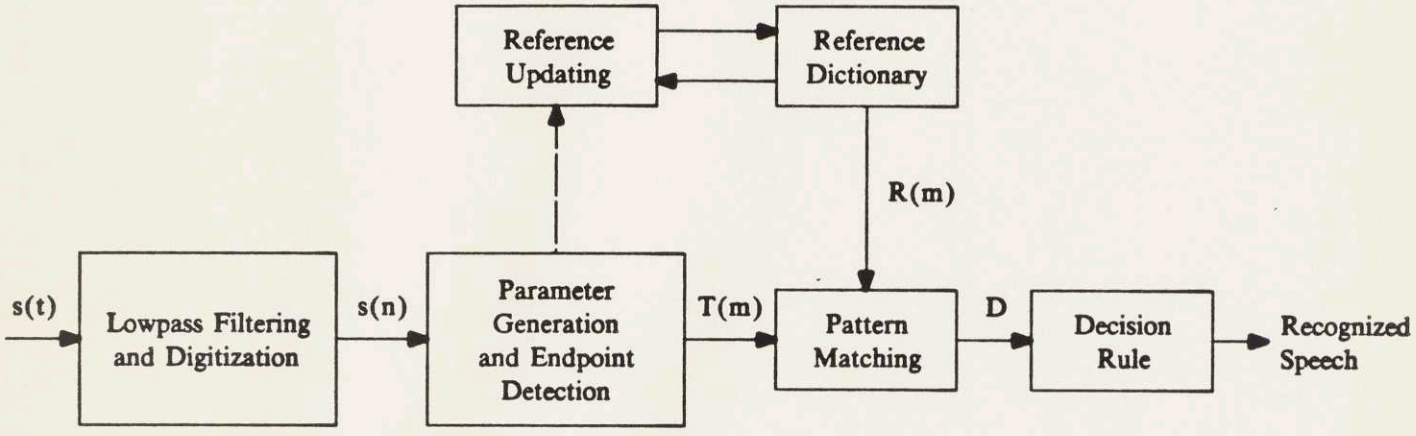


Figure 3.1. Isolated Word Recognition System (after Myers et.al.)

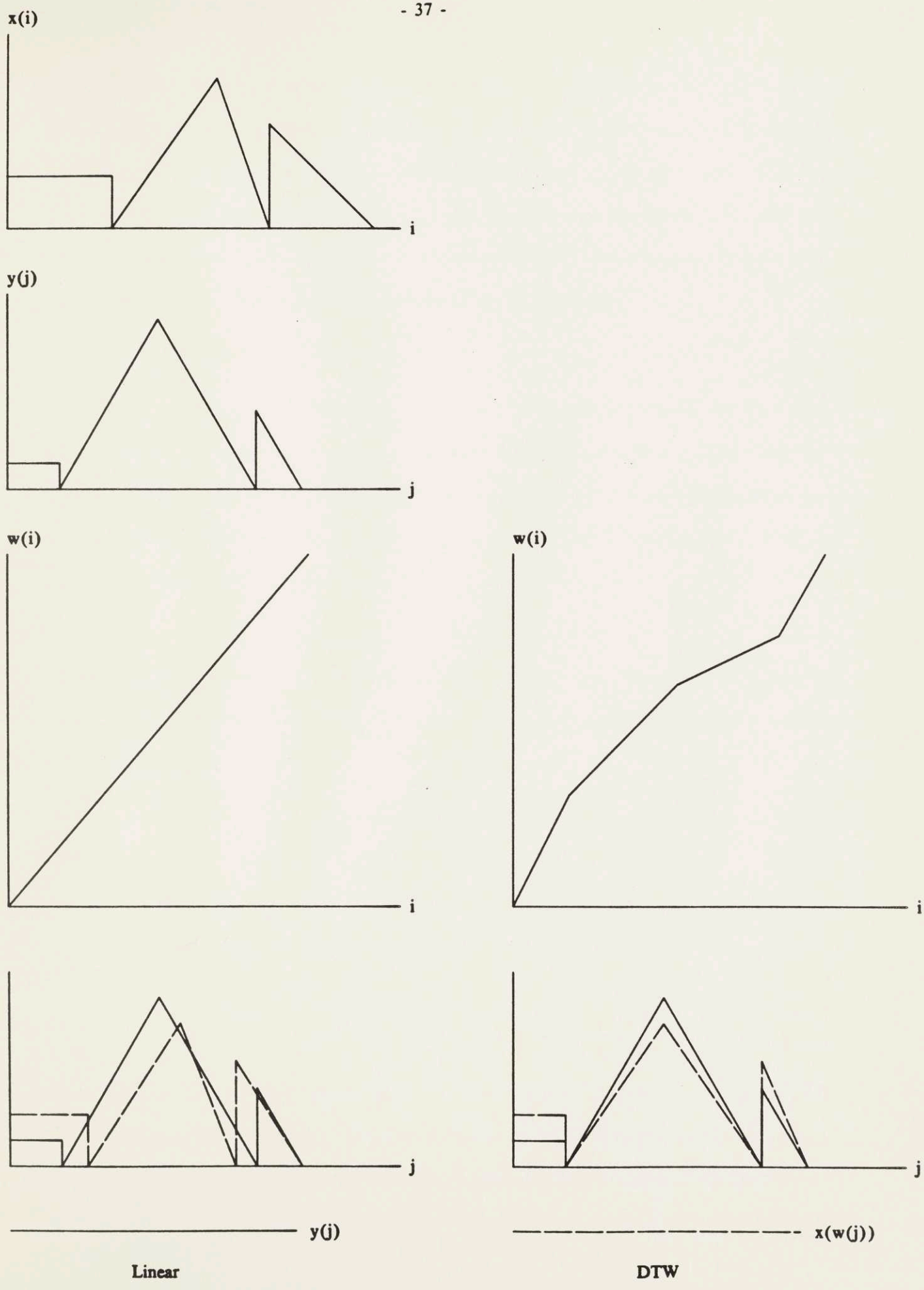


Figure 3.2. Comparison of Linear to DTW matching

similarity defined over the parameter space. The resulting surface will have a minimum along the path of closest similarity between test and reference. This path defines the line of summation which will yield the minimum possible total accumulated distance along the surface. It is the objective of Dynamic Time Warping to determine this path and the corresponding minimum total distance. Myers[27][8] presents an excellent discussion of Dynamic Time Warping and uses a general formulation of the problem which will be covered here in some detail.

3.1 Principles of Dynamic Time Warping

In Myers work the time alignment is formulated as a path finding problem with the search space defined by the two time axes, n and m, as depicted in Figure 3.3, where N and M are the number of frames in the reference and test utterances respectively. At each point in this grid space there is an implied distance, $d(\mathbf{R}(n),\mathbf{T}(m))$, which is the local distance between the n^{th} frame of the reference and the m^{th} frame of the test. This distance,

$$d(\mathbf{R}(n),\mathbf{T}(m)), \quad 1 \leqslant n \leqslant N,\ 1 \leqslant m \leqslant M, \tag{3.1}$$

defines the surface of interest. In general the time alignment path is specified by a pair of parametric equations $i(k)$ and $j(k)$ which map the test and reference time axes onto a third common time axis k, i.e.

$$n = i(k), \quad k = 1,2,\ldots,K \tag{3.2a}$$

$$m = j(k), \quad k = 1,2,\ldots,K \tag{3.2b}$$

where K is the length of the common time axis. This mapping is shown in Figure 3.3. The objective of DTW is to minimize the total distance,

$$D = \frac{\sum_{k=1}^{K} d(\mathbf{R}(i(k)),\mathbf{T}(j(k))\,\tilde{W}(k)}{\mathbf{N}(\tilde{W})} \tag{3.3}$$

where functions $i(k)$ and $j(k)$ are as defined above, $\tilde{W}(k)$ is a weighting function, and $\mathbf{N}(\tilde{W})$ is a normalization factor. In order to determine the minimum total distance along the surface defined over the (n, m) plane, the DTW algorithm places several constraints on the time alignment process. These are: endpoint constraints, local continuity constraints, and global path

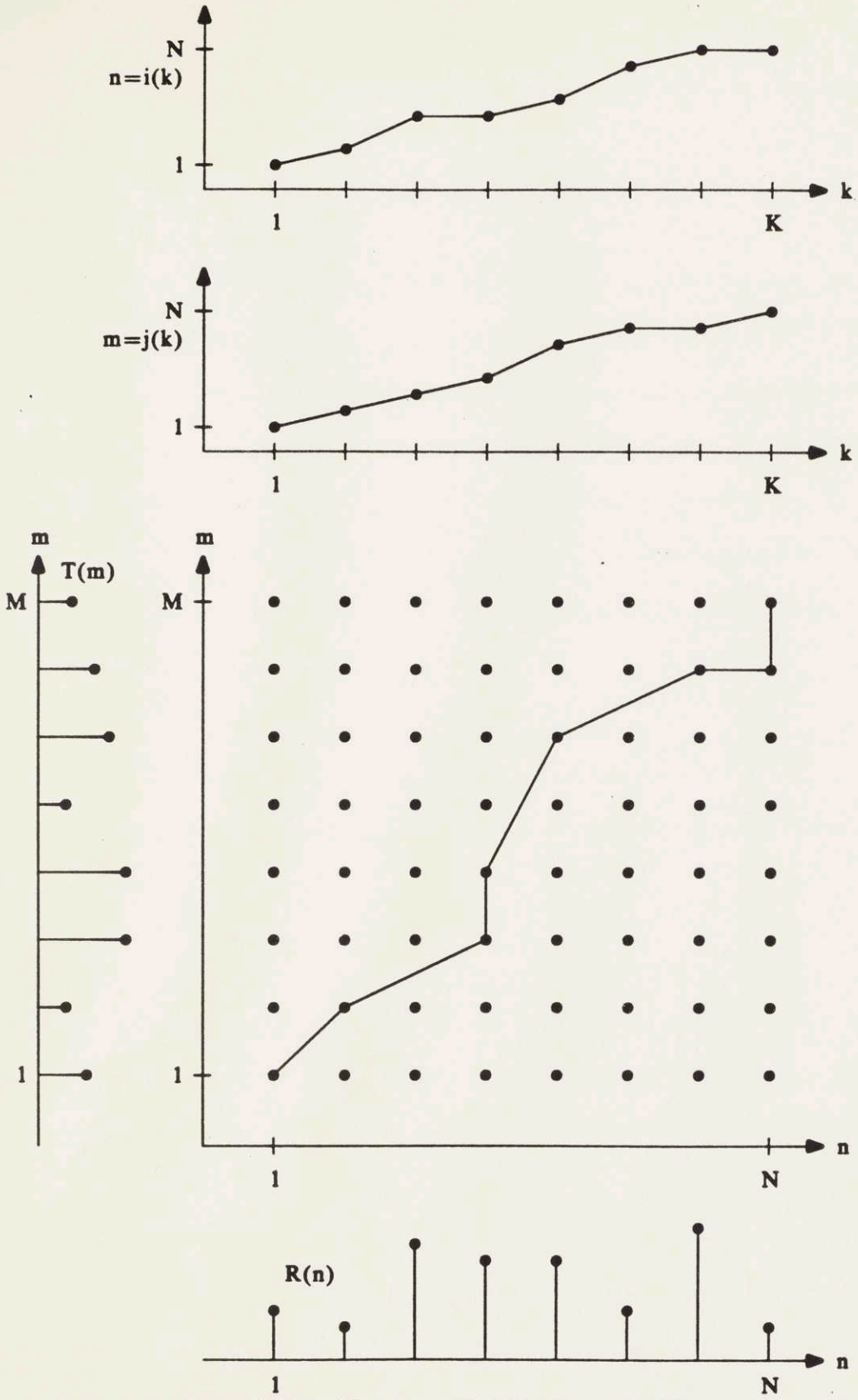


Figure 3.3. Dynamic Time Warping as Path Finding (after Myers et.al.)

constraints. These constraints and the means of computing the minimum total distance are covered in the following sections.

3.1.1 Endpoint Constraints

In isolated word recognition it is usually assumed that the beginning and ending of an utterance are well defined. Techniques for endpoint detection have been the subject of other work[28] [29] [30] and will not be discussed here. Given that the beginning and ending frames of each utterance are known one would like the time alignment to map the beginning of the test utterance to the beginning of the reference and similarly map the end of the test to the end of the reference. The time alignment path is therefore constrained to start at the point $(1,1)$, the first frames of the test and reference utterances, and end at the point (N, M), the final frames of the two utterances. This requirement gives the endpoint constraints of:

$$i(1) = 1,\ j(1) = 1 \qquad \textit{beginning point} \tag{3.4a}$$

$$i(K) = N,\ j(K) = M \qquad \textit{ending point.} \tag{3.4b}$$

The path shown in Figure 3.3 demonstrates this constraint, it starts at the point $(1,1)$ in the (n, m) plane and ends at the point (N, M).

3.1.2 Local Continuity Constraints

In order to prevent totally unrealistic time normalization, for example excessive local or overall compression or expansion of the time axes, some restrictions must be placed on the local variation that may occur in time alignment path. The first of these restrictions is that the alignment path be monotonically increasing, i.e.

$$i(k+1) \geqslant i(k) \tag{3.5a}$$

$$j(k+1) \geqslant j(k). \tag{3.5b}$$

This prevents the alignment path from going in the negative direction in either time dimension and insures that the time order of the two utterances will be maintained. The second restriction is the continuity or slope constraint which limits the amount of local compression or expansion that may occur by limiting the maximum and minimum slope of the alignment path. The slope constraints

can be written as

$$E_{MIN} \leq \frac{(j(k)-j(k-1))}{(i(k)-i(k-1))} \leq E_{MAX}, \quad k=1,2,3,\ldots,K \tag{3.6}$$

where E_{MAX} and E_{MIN} are the maximum and minimum slope constraints respectively. Usually the minimum slope is the reciprocal of the maximum slope constraint, i.e. $E_{MIN}=1/E_{MAX}$. Sakoe and Chiba[9] explored the effect of varying the slope constraints on recognition accuracy. They demonstrated that a maximum slope constraint of $E_{MAX}=2$ in the local path was the optimum choice.

These restrictions are termed the local continuity constraints. The local continuity constraints are given as simple local paths which are combined to make up the overall path. There are many possible local paths which can be used, Myers et.al.[8] explored the relative performance of several local continuity constraints. These local constraints are shown in Figure 3.4. Myers et.al. showed that the type I local constraints performed best, although not substantially better than the type II or III constraints. Figure 3.5 shows a sample path using type I local continuity constraints. The slope constraints which specify the maximum and minimum slope of the local paths also specify the maximum and minimum slope of the overall path, since the overall path is made up of a combination of the smaller local paths. Notice that the local constraints of type I, II, and III all have a maximum slope of two and a minimum slope of one half. The type I local continuity constraints were used in our system.

3.1.3 Global Path Constraints

The local continuity constraints will limit the time alignment path to lie within a restricted parallelogram shaped region of the total grid space. The limits of this parallelogram are defined by the inequalities,

$$1+\frac{(i(k)-1)}{E_{MAX}} \leq j(k) \leq 1+E_{MAX}(i(k)-1) \tag{3.7a}$$

$$M+E_{MAX}(i(k)-N) \leq j(k) \leq M+\frac{(i(k)-N)}{E_{MAX}} \tag{3.7b}$$

where E_{MAX} is the maximum slope constraint. This assumes that the minimum slope constraint is given by $E_{MIN} = 1/E_{MAX}$. Equation 3.7a represents the valid range of points which can be

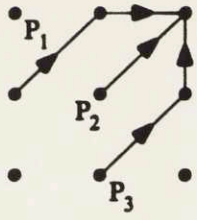
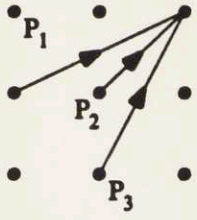
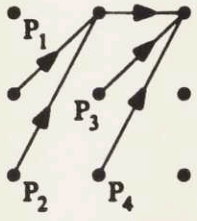
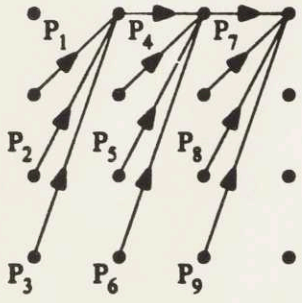
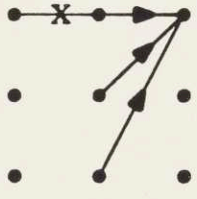
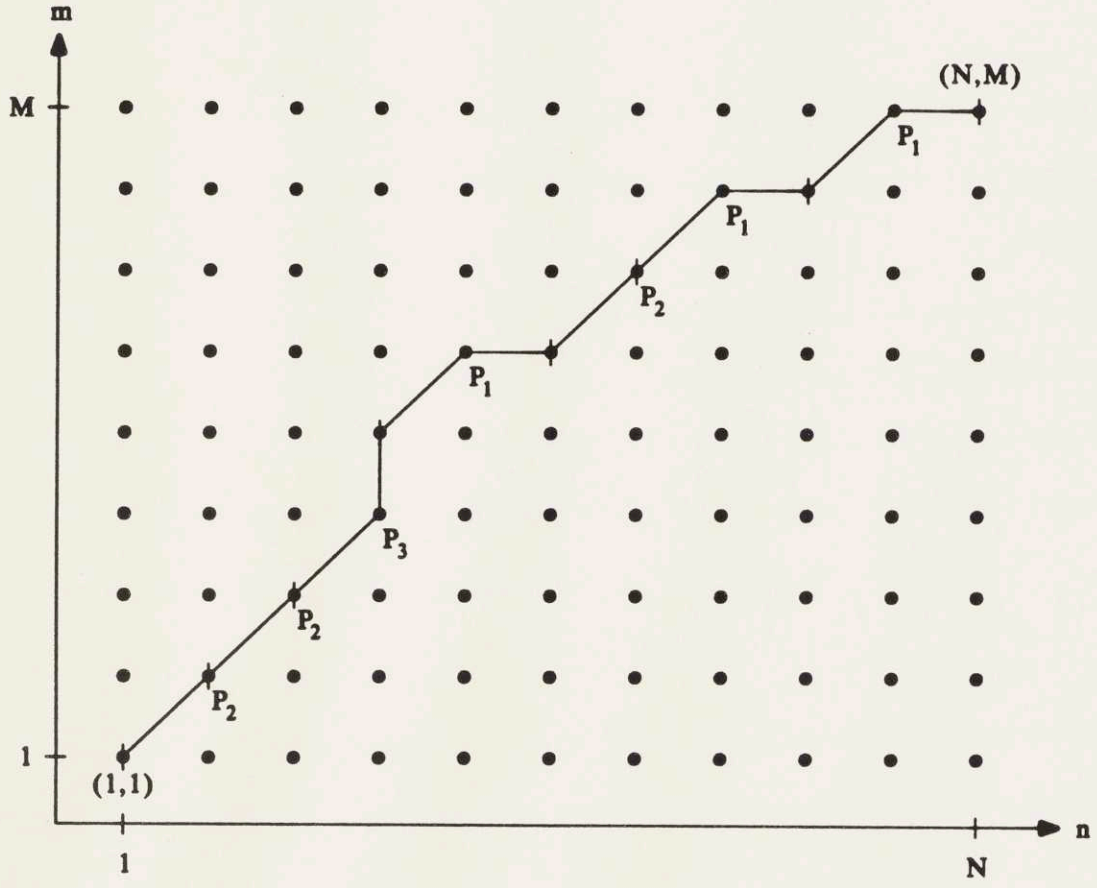
TYPE	PICTORIAL	PRODUCTIONS	E_{MAX}	E_{MIN}
I		$P_1 \rightarrow (1,0)(1,1)$ $P_2 \rightarrow (1,1)$ $P_3 \rightarrow (0,1)(1,1)$	2	1/2
II		$P_1 \rightarrow (2,1)$ $P_2 \rightarrow (1,1)$ $P_3 \rightarrow (1,2)$	2	1/2
III		$P_1 \rightarrow (1,0)(1,1)$ $P_2 \rightarrow (1,0)(1,2)$ $P_3 \rightarrow (1,1)$ $P_4 \rightarrow (1,2)$	2	1/2
IV		$P_1 \rightarrow (1,0)(1,0)(1,1)$ $P_2 \rightarrow (1,0)(1,0)(1,2)$ $P_3 \rightarrow (1,0)(1,0)(1,3)$ $P_4 \rightarrow (1,0)(1,1)$ $P_5 \rightarrow (1,0)(1,2)$ $P_6 \rightarrow (1,0)(1,3)$ $P_7 \rightarrow (1,1)$ $P_8 \rightarrow (1,2)$ $P_9 \rightarrow (1,3)$	3	1/3
IKATURA		NO PRODUCTION RULE CHARACTERIZATION	2	1/2

Figure 3.4. Local Continuity Constraints (after Myers et.al.)



SAMPLE PATH (TYPE I CONSTRAINTS)
PRODUCTIONS: $P_1\ P_1\ P_2\ P_1\ P_3\ P_2\ P_2$
PATH: (1,0)(1,1)(1,0)(1,1)(1,1)(1,0)(1,1)(0,1)(1,1)(1,1)(1,1)

Figure 3.5. Sample Path (after Myers et.al.)

reached from the beginning point (1,1). Equation 3.7b represents the valid range of points from which the ending point, (N, M), can be reached. There is an additional constraint on the overall path that was originally proposed by Sakoe and Chiba[9]. This constraint places an absolute limit on the global range of the overall path. This is called the absolute time difference range constraint and is written:

$$|i(k)-j(k)| \leqslant R \tag{3.8}$$

This constrains the absolute time displacement between frames in test and frames in the reference to be no more than RL/F_s seconds, where F_s is the sampling frequency and L is the number of samples between analysis intervals in the parameter generation. These global range constraints are shown in Figure 3.6. The absolute time difference range constraint was originally proposed as a method of reducing the amount of computation required in the DTW algorithm. Myers et.al.[8] showed that making use of this constraint reduced recognition accuracy for the normal DTW algorithm and for this reason no absolute range constraint was used in our system, i.e. $R=\infty$.

3.1.4 The Distance Function

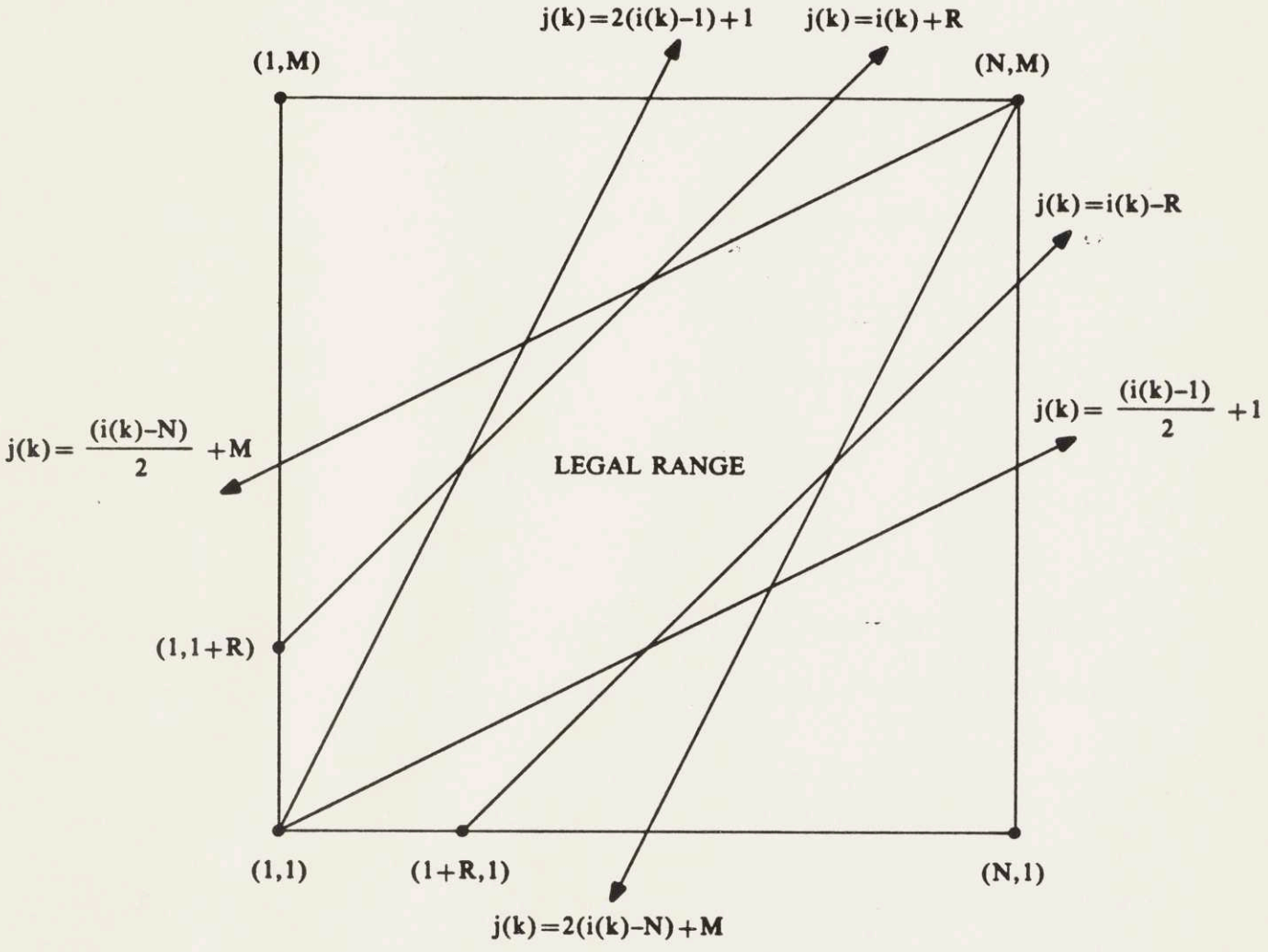
In determining the minimum total distance between a test and reference utterance the total distance along any path $(i(k), j(k))$ is written as a normalized weighted sum of the local distances $\tilde{d}(i(k), j(k))$

$$D(i(k), j(k)) = \frac{\sum_{k=1}^{K} \tilde{d}(i(k), j(k))\, \tilde{W}(k)}{\mathbf{N}(\tilde{W})} \tag{3.9}$$

where $\tilde{W}(k)$ is a weighting function and $\mathbf{N}(\tilde{W})$ is the normalization factor. For the DTW algorithms discussed by Myers the local distance is given by

$$\tilde{d}(n, m) = d(\mathbf{R}(n), \mathbf{T}(m)). \tag{3.10}$$

We shall see later how the local distance can be modified to include duration in the segmented case. To completely specify the distance function we must define $\tilde{W}(k)$ and $\mathbf{N}(\tilde{W})$. Typically the normalization factor will be dependent on the weighting function. Myers et.al.[8] present four potential weighting functions to be used in DTW and demonstrated that there was relatively little



$$E_{max} = 2$$

$$|N-M| \leq R$$

Figure 3.6. Global Range Constraints (after Myers et.al.)

change in recognition accuracy with the choice of weighting function. We chose a symmetric weighting referred to by Myers as type d which is written as

$$\tilde{W}(k) = i(k) - i(k-1) + j(k) - j(k-1) \tag{3.11}$$

The choice of $\mathbf{N}(\tilde{W})$ is usually made so that $D(i(k), j(k))$ is the average local distance along the path defined by $i(k)$ and $j(k)$. A natural choice for the normalization factor is the sum

$$\mathbf{N}(\tilde{W}) = \sum_{k=1}^{K} \tilde{W}(k). \tag{3.12}$$

For the weighting function chosen the normalization factor can be easily computed and is independent of the path, i.e.

$$\begin{aligned} \mathbf{N}(\tilde{W}) &= \sum_{k=1}^{K} (i(k)-i(k-1)+j(k)-j(k-1)) \qquad (3.13) \\ &= i(K)-i(0)+j(K)-j(0) \\ &= N+M. \end{aligned}$$

Now that we have covered the constraints placed on the time alignment path and the distance function which is to be minimized we will discuss how Dynamic Programming is used to determine the optimum path.

3.1.5 The Optimum Path

The best path is the set of functions $i(k)$, $j(k)$, $k=1,2,\ldots,K$ which minimizes the distance function of Equation 3.9. To arrive at this path a Dynamic Programming approach is used. The two principles on which this approach is based are[8]

1. a globally optimal path is also locally optimal,
2. the optimal path to the point (n,m) depends only on the values of n', m' such that $n' \leqslant n$, $m' \leqslant m$.

These two principles give rise to a recursive relationship which is the general form for Dynamic Programming. Consider the partially accumulated distance function

$$D_A(n,m) = \min_{(i(k),j(k),K')} \left[\sum_{k=1}^{K'} \tilde{d}(i(k),j(k))\, \tilde{W}(k) \right] \tag{3.14}$$

where $i(K')=n$, $j(K')=m$. Using one of the local continuity constraints discussed in the previous section we can write a simple recursive relationship for the partially accumulated distance. For example, using the type I local constraints and type d weighting function which were chosen for our system we have

$$D_A(n,m)=\min \begin{bmatrix} D_A(n-1,m-1)+2\tilde{d}(n,m) \\ D_A(n-1,m-2)+\frac{3}{2}(\tilde{d}(n,m-1)+\tilde{d}(n,m)) \\ D_A(n-2,m-1)+\frac{3}{2}(\tilde{d}(n-1,m)+\tilde{d}(n,m)) \end{bmatrix} \tag{3.15}$$

The solution for the total accumulated distance then simply becomes

$$\begin{aligned} D &= \frac{D_A(N,M)}{\mathbf{N}(\tilde{W})} \\ &= \frac{D_A(N,M)}{(N+M)}. \end{aligned} \tag{3.16}$$

and the DTW algorithm can be implemented in a three step procedure.

1. *Initialization:* Set $D_A(1,1) = \tilde{d}(1,1)\,\tilde{W}(1)$.

2. *Recursion:* Compute $D_A(n,m)$ recursively for $1 \leqslant n \leqslant N$, $1 \leqslant m \leqslant M$.

3. *Termination:* Set $D = D_A(N,M)/\mathbf{N}(\tilde{W})$.

Now that the basic principles of DTW have been discussed it is necessary to consider how the durational information contained in the segmented templates can be incorporated into the DTW algorithm.

3.2 Making Use of Durational Information

There are several possible ways to deal with the durational information contained in the segmented test and reference templates. The segment durations are an important factor in matching two templates, since the objective of Dynamic Time Warping is to determine the optimum time alignment of the two utterances. One of the major objectives of this thesis is to determine the

tradeoffs between recognition accuracy and computation time. The method in which the segmental durations are incorporated into the DTW algorithm will significantly affect performance in both of these areas. In this section we propose three alternative methods for handling durational information and discuss the implications of each in terms of recognition accuracy and computation.

It is difficult to determine how the nonlinear compression that is introduced by the segmentation will affect the ability of the DTW algorithm to align the utterances, since DTW is itself a nonlinear operation. It can be argued that by ignoring the segment durations and performing the time alignment directly on the segmented templates themselves, one can obtain better recognition accuracy than if this information were incorporated into the solution. For example, many of the errors that occur with the English alpha-digit vocabulary are mismatches among the words in the "B", "D", "G", etc. confusion set. These errors occur because more emphasis is placed on the steady state vowel portion than on the consonantal portion by virtue of the significantly longer duration of the vowel. One could view small differences in the steady state of the vowels in the test and reference utterances as noise introduced into the time alignment process which, if accumulated, could swamp the more significant transitional information. Acoustic segmentation emphasizes the changes in the spectral characteristics of an utterance giving more weight to the transitional regions, potentially resulting in better recognition accuracy.

An alternative argument is that by ignoring the durational information it is more difficult for the Dynamic Time Warping algorithm to align the two utterances. Since a nonlinear compression has already been performed on each of the utterances the required time warping function could exceed the slope or range constraints. The global range constraints may have to be relaxed to allow the time warping algorithm to compensate for the nonlinearities introduced by the segmentation. The range constraints may have to be relaxed so much as to allow totally unreasonable time alignments. In order to prevent this the durational information must be incorporated into the solution. Another argument for this case is that one would expect better performance to be achieved by taking advantage of all possible information, ignoring the durational information could be throwing away valuable data.

In specifying a DTW algorithm for segmented speech templates there are three factors which must be given additional consideration. These factors are the local distance, $\tilde{d}(n,m)$, the normalization factor, $\mathbf{N}(\tilde{W})$, and the effective size of the grid space, which is given by the expression for the minimum total distance, D. By specifying these three factors for segmented templates different methods for handling durational information can be implemented. In our work three methods of handling durational information were considered. The DTW algorithms which implement these methods will be discussed next.

3.2.1 Method 1: Ignoring Duration

In this method we take the approach that the best recognition accuracy can be obtained by ignoring durational information. Ignoring segment duration reduces the problem to one of normal Dynamic Time Warping using the representative parameter vectors of the segmented templates. This method is specified by the local distance

$$\tilde{d}(n,m) = d(\tilde{\mathbf{R}}(n),\tilde{\mathbf{T}}(m)), \tag{3.17a}$$

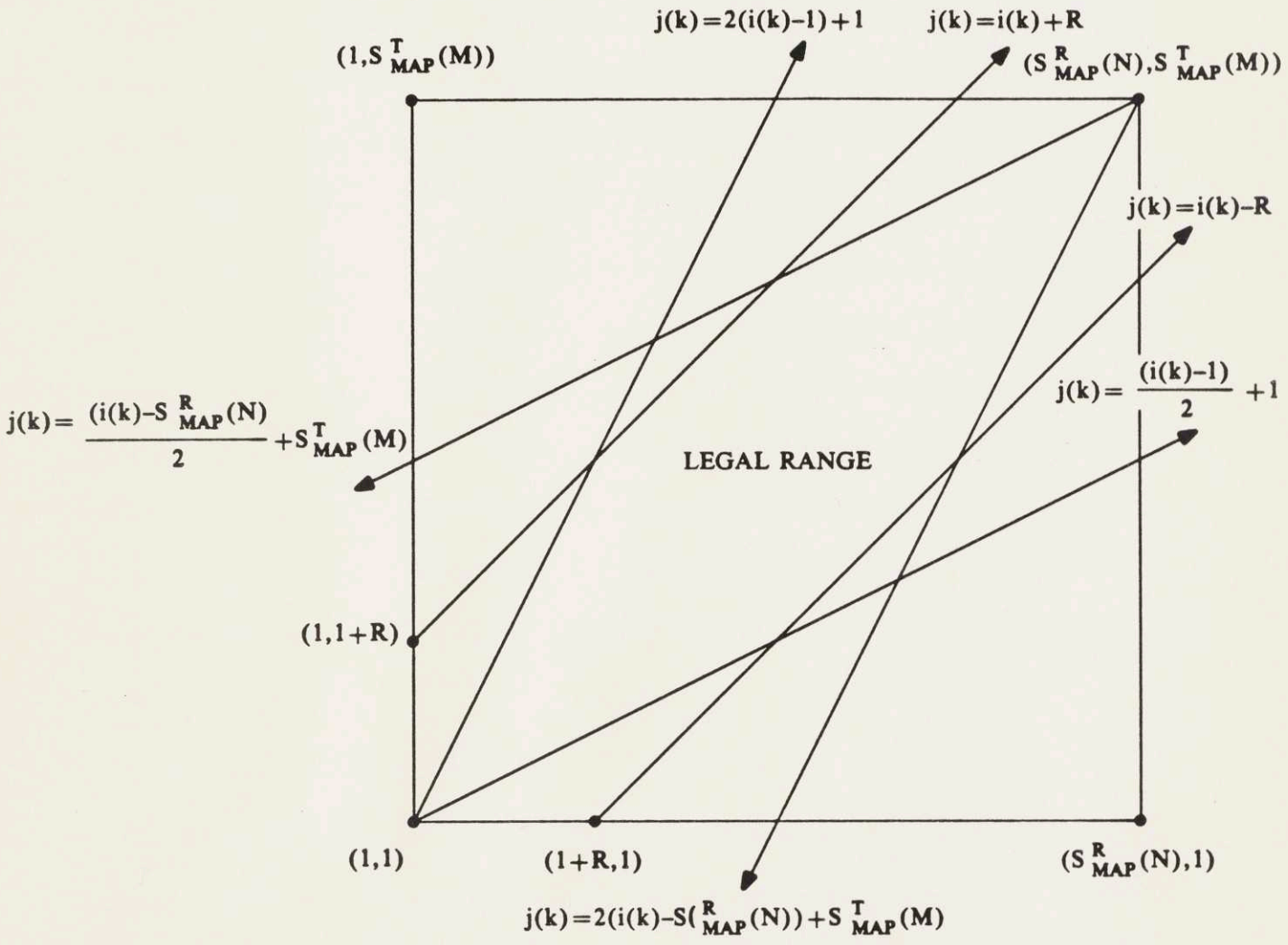
the normalization factor

$$\mathbf{N}(\tilde{W}) = S^R_{MAP}(N) + S^T_{MAP}(M), \tag{3.17b}$$

and the minimum total distance

$$D = \frac{D_A(S^R_{MAP}(N), S^T_{MAP}(M))}{\mathbf{N}(\tilde{W})}, \tag{3.17c}$$

where S^R_{MAP} and S^T_{MAP} are the segmental mapping functions for the segmented reference and test templates respectively. Since the segmental mapping function relates the frame number to the segment number, the number of segments in a segmented template is given by $S_{MAP}(N)$, where N is the number of frames in the utterance. The resulting grid space and the legal range for the segmented templates is shown in Figure 3.7. We see that the global range constraints and the overall size of the grid space are now defined in terms of the number of segments in the test and reference as opposed to the number of frames. The formulation of this method specifies that the local distances are the distances between the representative vectors of the segmented test and reference tem-



$$E_{max}=2$$

$$|N-M|\leq R$$

Figure 3.7. Grid Space for Segmented Templates

plates, as opposed to parameter vectors for a single frame in unsegmented case. The same local continuity constraints, global range constraints, and weighting functions can be used as in the unsegmented case since they do not influence the method of handling durational information. Using the weighting function discussed in section 3.1.4 and the reduced size of the grid space the normalization factor is given by the sum of the number of segments in the test and reference templates. This method is no different from the normal DTW algorithm excepting that it operates on a smaller grid space and uses the averaged segment vectors.

This approach to handling durational information promises to yield the greatest savings in computation since the number of local distances that need to be computed has been significantly reduced and the durational component is not used. If the segmentation algorithm is set to yield a data compression of 50% one can expect the average size of the grid space for the comparison of two words to be reduced by a factor of four. The relationship between compression and grid size is multiplicative because the size of the grid space is given by the product of the template lengths. To make use of this approach it is necessary that both the test and reference templates be of the segmented form.

3.2.2 Method 2: Weighting by Average

Determining the best means of incorporating the durational information into the Dynamic Programming solution is a difficult problem since there are many alternatives available. A simple approach is to weight the local distances between segment vectors by a function, G, of the segment lengths, i.e.

$$\tilde{d}(n,m) = G(L_R(n), L_T(m)) \cdot d(\tilde{\mathbf{R}}(n), \tilde{\mathbf{T}}(m)) \tag{3.18}$$

where $L_R(n)$ and $L_T(m)$ are the length, in number of frames, of the n^{th} segment of the reference and the m^{th} segment of the test respectively. This approach increases the amount of computation over the previous method only slightly, yet it incorporates the durational information into the solution.

Many alternative weighting functions could be suggested, such as the sum or product of the segment lengths, or perhaps a more complicated weighting function. The weighting function should reflect the contribution of the local distance to total accumulated distance. The weighting function type d discussed in section 3.1.4 uses the sum of the change in the n and m directions to weight the local distances. To emulate this weighting function the weighting should be given by the sum of the segment lengths. However, this weighting function must be normalized to give unity for the case when both the test and reference segments are one frame in length. Consequently it was decided that a simple average of the lengths of the two segments should be used to weight the local distance computed between the representative vectors, i.e.

$$G(L_R(n), L_T(m)) = \frac{L_R(n) + L_T(m)}{2} \tag{3.19}$$

The DTW algorithm for this method is specified as follows, the local distances are given by

$$\tilde{d}(n, m) = \frac{L_R(n) + L_T(m)}{2} \cdot d(\tilde{\mathbf{R}}(n), \tilde{\mathbf{T}}(m)), \tag{3.20a}$$

the normalization factor by

$$\mathbf{N}(\tilde{W}) = N + M, \tag{3.20b}$$

and the total accumulated distance is

$$D = \frac{D_A(S^R_{MAP}(N), S^T_{MAP}(M))}{\mathbf{N}(\tilde{W})}. \tag{3.20c}$$

Notice that the normalization factor is the sum of the number of frames instead of the number of segments as in the previous method, this is to account for the weighting of the local distances by the average segment length. The grid space for this method is identical to the one for Method 1 as the only changes made to the algorithm where in the local distance and the normalization factor.

3.2.3 Method 3: Expanding Segments

Another approach to handling durational information is to use this information in such a way as to relate the solution for segmented templates to the solution for the unsegmented form. Previous work has demonstrated that the solution for the unsegmented form gives excellent results for many

vocabularies. Conceptually the simplest means of relating the segmented form to the original unsegmented form is to expand the segments to their original length by repeating the average frame the appropriate number of times. One could think of this as creating a new set of speech templates as follows

$$\tilde{\mathbf{R}}'(n) = \tilde{\mathbf{R}}(S^R_{MAP}(n)), \qquad n=1,2,\ldots,N \tag{3.21a}$$

$$\tilde{\mathbf{T}}'(m) = \tilde{\mathbf{T}}(S^T_{MAP}(m)), \qquad m=1,2,\ldots,M \tag{3.21b}$$

and using the normal DTW algorithm on these new templates. The resulting grid space will be made up of areas of constant distance where segments intersect, the distance between the representative vectors being repeated for each point in such an area, as illustrated in Figure 3.8. Note that the size of the grid space has been increased to the original unsegmented case, i.e. the product of the number of frames in the test and reference. The global range constraints are again specified in terms of the number of frames as in the unsegmented case. As in the previous methods this method reduces the amount of computation required for the local distances, since each rectangular area in Figure 3.8 requires the distance be computed only once. However, the size of the grid space has been increased back to the original size and the number of paths which must be traced in the Dynamic Programming algorithm has increased correspondingly. This method will show improvement in the computational requirements over the standard Dynamic Time Warping algorithm, but such improvement will not be as great as the previous two methods. By taking advantage of the segmental mapping functions created by the segmentation system the minimum total distance, D, can be efficiently evaluated using the following specifications on the DTW algorithm, the local distance is given by

$$\tilde{d}(n,m) = d(\tilde{\mathbf{R}}(S^R_{MAP}(n)),\tilde{\mathbf{T}}(S^T_{MAP}(m))), \tag{3.22a}$$

the normalization factor by

$$\mathbf{N}(\tilde{W}) = N + M, \tag{3.22b}$$

and the minimum total distance is given by

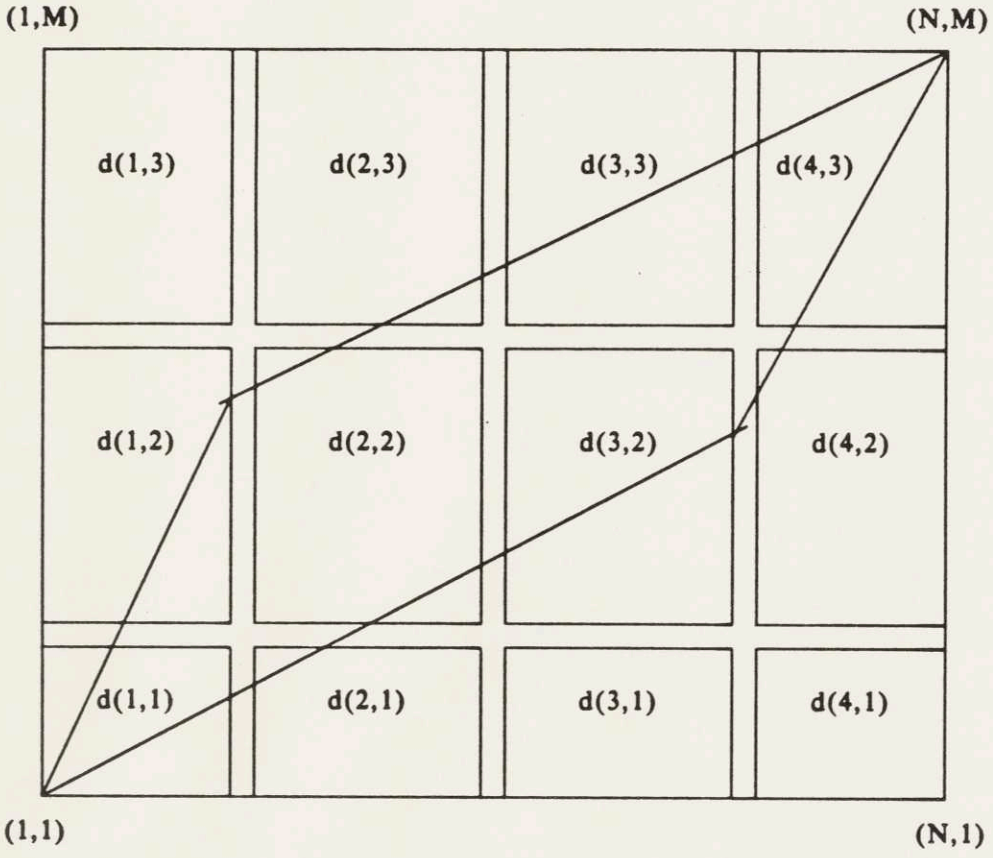


Figure 3.8. The Grid Space With Expanded Segments

$$D = \frac{D_A(N, M)}{\mathrm{N}(\bar{W})}. \tag{3.22b}$$

The segmental mapping functions are used in Equation (3.22a) to map the frame number to the segment number which correspond to the rectangular area in which the frame is located. The normalization factor is the sum $N+M$ as the effective path length has been expanded to original unsegmented case.

3.3 Summary

We have discussed the basic principles of Dynamic Time Warping and the factors involved in specifying a DTW algorithm. The specifications for the DTW algorithm used in our experiments are shown in Table 3.1. Implications for the DTW algorithm using segmented speech templates were discussed, three methods for making use of durational information were proposed, and specifications in the mathematical formalism of Myers et.al.[8] were given. Table 3.2 shows the specifications for the DTW algorithm for the unsegmented case and for the three methods of handling durational information for the segmented case. These three methods of handling durational information were compared to determine the relative performance in terms of recognition accuracy and computational requirements. These experiments and their results will be covered in chapter five. In the next chapter we will discuss the parameter sets used and some of their properties as well as develop the distance measures that are to be contrasted.

Endpoint Constraints	$i(1)=1,\ j(1)=1$ $i(K)=N,\ j(K)=M$
Local Continuity Constraints	$D_A(n,m)=\min \begin{cases} D_A(n-1,m-1)+2\tilde{d}(n,m) \\ D_A(n-1,m-2)+\frac{3}{2}(\tilde{d}(n,m-1)+\tilde{d}(n,m)) \\ D_A(n-2,m-1)+\frac{3}{2}(\tilde{d}(n-1,m)+\tilde{d}(n,m)) \end{cases}$
Global Range Constraints	$1+\frac{(i(k)-1)}{2} < j(k) < 1+2(i(k)-1)$ $M+2(i(k)-N) < j(k) < M+\frac{(i(k)-N)}{2}$ $R = \infty$

TABLE 3.1. Specifications for DTW Algorithm

Method	Local Distance	Normalization Factor	Minimum Total Distance
Normal	$\tilde{d}(n,m)=d(\mathbf{R}(n),\mathbf{T}(m))$	$\mathbf{N}(\tilde{W})=N+M$	$D=\frac{D_A(N,M)}{\mathbf{N}(\tilde{W})}$
Method 1	$\tilde{d}(n,m)=d(\tilde{\mathbf{R}}(n),\tilde{\mathbf{T}}(m))$	$\mathbf{N}(\tilde{W})=S^R_{MAP}(N)+S^T_{MAP}(M)$	$D=\frac{D_A(S^R_{MAP}(N),S^T_{MAP}(M))}{\mathbf{N}(\tilde{W})}$
Method 2	$\tilde{d}(n,m)=\frac{L_R(n)+L_T(m)}{2}\cdot d(\tilde{\mathbf{R}}(n),\tilde{\mathbf{T}}(m))$	$\mathbf{N}(\tilde{W})=N+M$	$D=\frac{D_A(S^R_{MAP}(N),S^T_{MAP}(M))}{\mathbf{N}(\tilde{W})}$
Method 3	$\tilde{d}(n,m)=d(\tilde{\mathbf{R}}(S^R_{MAP}(n)),\tilde{\mathbf{T}}(S^T_{MAP}(m)))$	$\mathbf{N}(\tilde{W})=N+M$	$D=\frac{D_A(N,M)}{\mathbf{N}(\tilde{W})}$

TABLE 3.2. Specifications for DTW Algorithm, Segmented Case

4. SPEECH PARAMETERS

While there has been considerable research done in the area of speech parameters, it has been very difficult to determine the cues and features of the speech signal which are important to human speech perception and understanding. To date there is no complete and conclusive agreement among speech scientists as to a specific set of speech parameters. However, there is some general agreement as to some of the more fundamental ones. For example, the formants or resonant peaks of the vocal tract transfer function, and the general spectral shape are unquestionably of critical importance. Precisely what cues are taken from this information is not fully understood. But one can take the approach that an accurate spectral representation is necessary for a speech recognition system.

Representation of the entire spectrum of a frame of speech data, such as by a DFT, poses some difficult problems. Among other reasons, such as glottal influence and harmonics of fundamental frequency, the amount of data involved is considerable. The number of useful spectral points is one half the DFT length and is often dictated by the frequency resolution desired. In general a method of spectral estimation or smoothing which is specified by a relatively small set of parameters, such as Linear Predictive Coding (LPC), or a Cepstral representation, is preferred.

4.1 Linear Prediction and LPC Parameters

Linear prediction is a method of spectral estimation that is based upon a simple all-pole model of the sequence of interest, i.e. the z-transform of the sequence is assumed to be of the form

$$H(z) = \frac{\sigma}{1 - \sum_{k=1}^{p} a_k z^{-k}} = \frac{\sigma}{A(z)}. \tag{4.1}$$

The basic theory underlying linear prediction, or linear predictive coding, is covered in great detail by Rabiner and Schafer[21] and by Markel and Gray[31]. We will only briefly summarize this work and give the method of solution used in our work.

Typically LPC analysis will be applied to a windowed portion of the speech signal,

$$x(m) = s(m+n)\,w(m) \tag{4.2}$$

where $w(m)$ is a finite length window which is defined over the interval $0 \leqslant m \leqslant N-1$ and is zero outside this interval. N is the number of samples contained in the window function. In order to generate the coefficients of the LPC model using the autocorrelation method it is necessary to generate the first $p+1$ autocorrelation coefficients, where p is the order of the model. The short time autocorrelation is defined as

$$R(k) = \sum_{m=0}^{N-1-k} x(m)\,x(m+k). \tag{4.3}$$

The major advantage of LPC analysis is that it makes use of a least squares error criterion which yields a simple set of linear equations which can be efficiently solved for the inverse filter coefficients, a_k. The set of equations yielded by the autocorrelation analysis can be written

$$\sum_{k=1}^{p} a_k R(|i-k|) = R(i), \quad i = 1,2,\ldots,p \tag{4.4}$$

and expressed in matrix form

$$\begin{bmatrix} R(0) & R(1) & R(2) & \cdots & R(p-1) \\ R(1) & R(0) & R(1) & \cdots & R(p-2) \\ R(2) & R(1) & R(0) & \cdots & R(p-3) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ R(p-1) & R(p-2) & R(p-3) & \cdots & R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \cdots \\ \cdots \\ a_p \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ R(3) \\ \cdots \\ \cdots \\ R(p) \end{bmatrix} \tag{4.5}$$

A result of using the autocorrelation method is that the matrix equation is of Toeplitz form, is guaranteed to be nonsingular, and lends itself to efficient methods of solution. One of the most efficient solutions to this type of matrix equation is Durbin's recursive solution[21]. This method is an iterative approach in which the $i+1^{st}$ solution is based on the i^{th} solution.

This recursion is performed as follows:

$$E^{(0)} = R(0) \tag{4.6a}$$

$$k_i = \frac{\left\{ R(i) - \sum_{j=1}^{i-1} a_j^{(i-1)} R(i-j) \right\}}{E^{(i-1)}} \qquad 1 \leqslant i \leqslant p \tag{4.6b}$$

$$a_j^{(i)} = k_i \tag{4.6c}$$

$$a_j^{(i)} = a_j^{(i-1)} - k_i a_{i-1}^{(i-1)} \qquad 1 \leqslant j \leqslant i-1 \tag{4.6d}$$

$$E^{(i)} = (1 - k_i) E^{(i-1)} \tag{4.6e}$$

Equations (4.6b)-(4.6e) are solved recursively for $i = 1,2,\ldots,p$ and the final solution is given as

$$\alpha = E^{(p)} \tag{4.7}$$

$$a_0 = -1$$

$$a_j = a_j^{(p)}, \quad 1 \leqslant j \leqslant p$$

where α is the total square error, also called the minimum residual error.

The minimum residual error, α, is the energy in the signal $e(n)$ that results when the windowed signal $s(n)$ is passed through the filter with z-transform $A(z)$, as shown in Figure 4.1. The minimal residual error is thus written as

$$\alpha = \sum_{n=0}^{N+p-1} e^2(n). \tag{4.8}$$

The filter $A(z)$ is known as the inverse filter or the for the sequence $x(n)$. The coefficients k_i are known as the partial correlation or PARCOR coefficients, and are related to reflection coefficients in an acoustic tube model of the vocal tract. One of the advantages of the PARCOR coefficients is that they can be linearly interpolated and are guaranteed to give a stable filter. We will take advantage of this fact when we discuss computation of the average parameter vectors.

4.2 LPC Derived Cepstral Parameters

Another parameter set that is frequently used in speech processing is the low order cepstral coefficients of the homomorphically smoothed spectra. The properties of homomorphic systems for

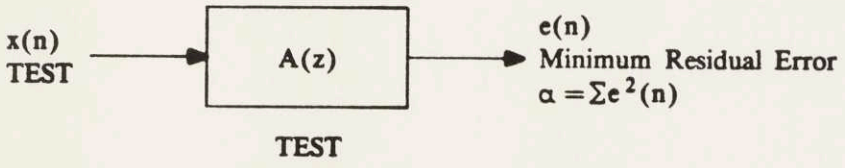


Figure 4.1. Minimum Residual Error From Inverse Filter

convolution are discussed in great detail by Oppenheim and Schafer[32]. The cepstrum is defined as the inverse Fourier transform of the log magnitude spectrum. Samples of the cepstrum are normally computed using the inverse discrete Fourier transform, i.e.

$$c_n = \frac{1}{N}\sum_{k=0}^{N-1} \ln|X(k)|\, e^{j\frac{2\pi}{N}kn}, \quad 0 \leqslant n \leqslant N-1 \tag{4.9}$$

where $X(k)$ is the discrete Fourier transform of the sequence $x(n)$. One of the disadvantages of computing the cepstrum directly from the speech signal itself is that it requires two FFT's and a logarithm to generate the cepstral coefficients. For an all-pole system, however, the conversion from an LPC spectral estimate to a cepstral representation turns out to be a fairly simple matter as we shall now see.

The Fourier series expansion for the magnitude squared of the LPC spectral estimate is given by.

$$\ln\left[\frac{\sigma^2}{|A(e^{j\theta})|^2}\right] = \sum_{k=-\infty}^{\infty} c_k e^{-jk\theta} \tag{4.10}$$

where

$$c_0 = \ln\left[\sigma^2\right] \tag{4.11}$$

and

$$c_{-k} = c_k. \tag{4.12}$$

Atal[33] presents an efficient method of converting from the inverse filter coefficients a_k to the cepstral coefficients c_k of the corresponding spectrum. The recursive relationship can be written as

$$c_1 = a_1 \tag{4.13a}$$

$$c_n = \sum_{k=1}^{n-1} (1-k/n)\, a_k\, c_{n-k}, \quad 1 < n < p \tag{4.13b}$$

$$c_n = \sum_{k=1}^{n-1} (1-k/n)\, a_k\, c_{n-k}, \quad n > p. \tag{4.13c}$$

By defining a new variable

$$g_n = nc_n \tag{4.14}$$

this equation can be rewritten in a simpler form:

$$g_1 = a_1 \tag{4.15a}$$

$$g_n = na_n + \sum_{k=1}^{n-1} a_k g_{n-k} \tag{4.15b}$$

where $a_n = 0$ for $n > p$. It should be emphasized that the resulting cepstrum is not what would be obtained directly from the sequence $x(n)$, but is derived from the LPC spectral estimate $\sigma / A(z)$.

4.3 Computing the Average Parameter Vector

As was discussed in section 2.2.6 better recognition accuracy could be expected if the segments were to be represented by the parameter vector which is the average of all the parameter vectors within the segment. The average in this case is the arithmetic mean of these vectors. Due to the nature of the parameters chosen computation of the average vector can not be achieved in a straight forward matter. The means of computing the average vector will now be discussed.

For LPC parameters the filter coefficients, a_k, cannot be linearly interpolated and still obtain a stable filter[21] The PARCOR coefficients, however, do not suffer from this restriction as they are directly related to the reflection coefficients of an acoustic tube model. Linear interpolation of the PARCOR coefficients is guaranteed to give a stable filter. The method of obtaining the average LPC parameter vector is to compute the average PARCOR coefficients by

$$\bar{k}_i = \frac{1}{(m_f - m_i + 1)} \sum_{m=m_i}^{m_f} k_i(m), \quad 1 \leqslant i \leqslant p \tag{4.16}$$

where m_i is the initial frame and m_f is the final frame of the segment. The average energy, $\bar{R}(0)$, the *zero*th autocorrelation coefficient of the average filter is computed as

$$\bar{R}(0) = \frac{1}{(m_f - m_i + 1)} \sum_{m=m_i}^{m_f} R_m(0), \tag{4.17}$$

where $R_m(0)$ is the energy of the m^{th} frame. The filter coefficients, $\bar{a}_i$, the autocorrelation

coefficients of the impulse response, $\bar{R}(i)$, and the minimum residual error, $\bar{\alpha}$, can be obtained for the averaged representation through the following relationships. The average filter coefficients may be obtained from the average PARCOR coefficients by the recursion

$$\bar{a}_i^{(i)} = \bar{k}_i \tag{4.18a}$$

$$\bar{a}_j^{(i)} = \bar{a}_j^{(i-1)} - \bar{k}_i \bar{a}_{i-j}^{(i-1)}, \quad 1 \leqslant j \leqslant i-1. \tag{4.18b}$$

The remaining autocorrelation coefficients for the average filter are computed using the following recursion

$$\bar{E}^{(0)} = \bar{R}(0) \tag{4.19a}$$

$$\bar{R}(i) = \bar{k}_i \bar{E}^{(i-1)} + \sum_{j=1}^{i-1} \bar{a}_j^{(i-1)} \bar{R}(i-j), \quad 1 \leqslant i \leqslant p \tag{4.19b}$$

$$\bar{E}^{(i)} = (1-\bar{k}_i^2)\, \bar{E}^{(i-1)} \tag{4.19c}$$

$$\bar{\alpha} = \bar{E}^{(p)} \tag{4.19d}$$

The average autocorrelation coefficients for the inverse filter polynomial can be obtained from the average filter coefficients as follows

$$\bar{b}(n) = \sum_{k=0}^{p-n} \bar{a}_k \bar{a}_{k+i} \tag{4.20}$$

The average parameter vector for the cepstral representation is much more straight forward, since the arithmetic average of the cepstral coefficients yields the cepstral coefficients of the arithmetic average of the log spectra.

$$\bar{c}_k = \frac{1}{(m_f - m_i + 1)} \cdot \sum_{m=m_i}^{m_f} c_k(m), \quad -\infty \leqslant k \leqslant \infty. \tag{4.21}$$

The range given for the subscript k in the above equation is minus infinity to plus infinity, however, only the first L+1 coefficients are used since the filter characteristics are contained in the low order coefficients. Taking advantage of this and the symmetry of the cepstral coefficients the range on k in Equation (4.21) can be limited to $0 \leqslant k \leqslant L$.

4.4 Distance Measures

Distance measures have many applications in the field of speech processing and considerable research has been done to investigate the properties of many different distance measures. A distance measure should give a value related to the similarity of the two items being considered, the closer these items are in their parameter space the smaller this value the farther apart the greater this value. When discussing distance measures it is normal to talk about distance metrics which are defined over the vector space. The properties which define a metric over a vector space are the following

$$d(\mathbf{x},\mathbf{y}) = d(\mathbf{y},\mathbf{x}) \tag{4.22a}$$

$$\begin{aligned} d(\mathbf{x},\mathbf{y}) &> 0 \quad \text{for } \mathbf{x}\neq\mathbf{y} \\ &= 0 \quad \text{for } \mathbf{x}=\mathbf{y} \end{aligned} \tag{4.22b}$$

$$d(\mathbf{x},\mathbf{y}) \leq d(\mathbf{x},\mathbf{z}) + d(\mathbf{y},\mathbf{z}). \tag{4.22c}$$

The property of symmetry is expressed by Equation (4.22a), positive definiteness by Equation (4.22b), and Equation (4.22c) is referred to as the triangle inequality. When speaking of distance it is normally assumed that the distance from $\mathbf{x}$ to $\mathbf{y}$ is equal to the distance from $\mathbf{y}$ to $\mathbf{x}$, and the distance between $\mathbf{x}$ and $\mathbf{y}$ is positive, except for the case where $\mathbf{x}=\mathbf{y}$ and the distance is zero. Often one or more of these properties will be not be satisfied in favor of factors more appropriate to speech processing, such as the efficiency with which $d(\mathbf{x},\mathbf{y})$ can be evaluated[18] or correlation of $d(\mathbf{x},\mathbf{y})$ to perceptual distance[25].

Consider two spectral models $\sigma / A(z)$ and $\sigma' / A'(z)$. The error between these two spectra is given by

$$V(\theta) = \ln\left[\sigma^2 / |A(e^{j\theta})|^2\right] - \ln\left[(\sigma')^2 / |A'(e^{j\theta})|^2\right] \tag{4.23}$$

One set of distance measures which might be considered is the set of L_p norms defined by

$$(d_p)^p = \int_{-\pi}^{\pi} |V(\theta)|^p \frac{d\theta}{2\pi}. \tag{4.24}$$

The mean absolute log spectral measure is given by $p=1$ and the rms log spectral measure is given for $p=2$. The L_p norms are related to the decibel scale by the multiplicative factor $10/\ln(10)=4.3429\ldots$. While the L_p norms are true metrics in that they satisfy all the requirements for a metric given by Equation (4.22), it is computationally expensive, requiring two FFT's and logarithms. As a result this method is prohibitive for most applications and other measures of acoustic similarity must be explored. In the following sections we will discuss three distance measures presented by Gray and Markel[18] which can be related to the L_2 norm and can be evaluated more efficiently. Note that for all the distance measures the distance between two average parameter vectors is computed in the same way.

4.4.1 Itakura's Log Likelihood Ratio

One of the distance measures discussed by Gray and Markel is the log likelihood ratio originally proposed by Itakura[7]. This distance measure is based on the ratio of the residual errors resulting from the two spectral models that are being compared. Consider the minimum residual error α for the sequence $x(n)$ and the inverse filter $A(z)$. If this sequence is passed through a different inverse filter $A'(z)$ which minimizes the error α' for another sequence $x'(n)$, see Figure 4.2, the resulting residual error, δ, must be greater than or equal to the minimum residual error α, i.e.

$$\delta = \sum_{n=0}^{N+p-1} \left[\sum_{i=0}^{p} a'_i x(n-i) \right]^2 \geq \alpha \tag{4.25}$$

with the equality holding if and only if $A(z) = A'(z)$. The ratio δ/α must always be greater than or equal to one. The greater the difference in the sequences $x(n)$ and $x'(n)$ and thus in the spectral models $\sigma/A(z)$ and $\sigma'/A'(z)$ the larger this ratio. This ratio provides a measure of acoustic similarity between the two spectral models. Typically this measure is given as the log of the ratio and is expressed in decibels by

$$d(\mathbf{x},\mathbf{x}') = 4.3429 \ln \left[\delta/\alpha \right] \tag{4.26}$$

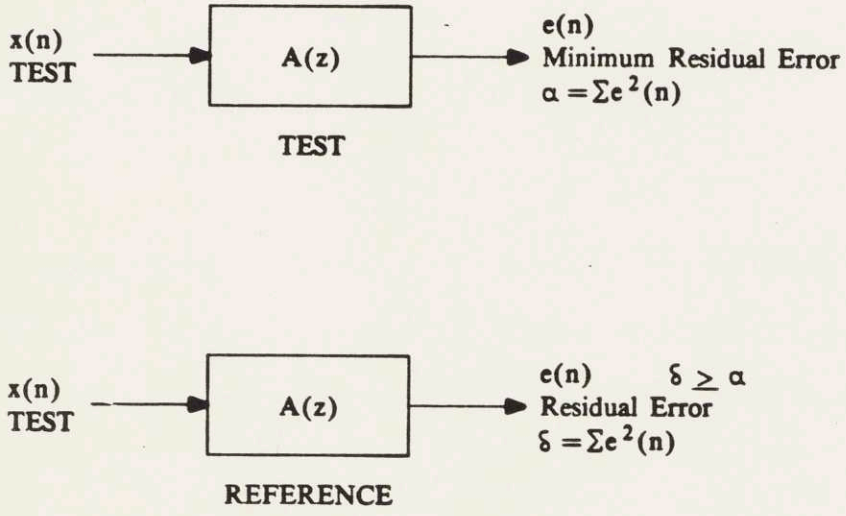


Figure 4.2. Minimum Residual and Residual Error From Inverse Filters

This measure is referred to as a log likelihood ratio as under certain assumptions on the data and window size it can be shown that this ratio is equivalent to a log likelihood ratio.

The log likelihood ratio can be efficiently computed through the use of autocorrelation sequences. The minimum residual error can be computed as follows

$$\alpha = \sum_{n=-p}^{p} b(n) R(n) \tag{4.27}$$

$$= b(0) R(0) + 2 \sum_{n=1}^{p} b(n) R(n),$$

where $b(n)$ is the autocorrelation of the inverse filter coefficients, a_k, i.e.

$$b(n) = \sum_{k=0}^{p-n} a_k a_{k+i} \tag{4.28}$$

and $R(n)$ is the autocorrelation of the sequence $x(n)$. Since the minimum residual error is a result of the LPC analysis it is normally saved as a component of the parameter vector and not recomputed. The residual error, however, must be computed and is similarly given by the relation

$$\delta = b'(0) R(0) + 2 \sum_{n=0}^{p} b'(n) R(n). \tag{4.29}$$

The log likelihood ratio is a nonlinear measure which gives more weighting to peaks in $V(\theta)$. Gray and Markel propose the following nonlinear relationship to approximate the L_2 norm from the likelihood ratio

$$d_2 \approx \sqrt{2(\delta/\alpha - 1)} \quad \text{for} \quad |V(\theta)| << 1. \tag{4.30}$$

This relationship was discarded in favor of using the distance measure as it is defined in Equation (4.26) in order to demonstrate the relative performance of this widely used distance measure.

4.4.2 The Cosh Distance Measure

One of the drawbacks to the log likelihood ratio is that it is not a symmetric measure, i.e.

$$d(\mathbf{x},\mathbf{x}') \neq d(\mathbf{x}',\mathbf{x}). \tag{4.31}$$

It is a desirable property of a distance measure in speech processing that the distance between two spectral models be independent of which model is chosen as the test and which is the reference.

Gray and Markel[18] propose a symmetric distance measure based on the average of two nonsymmetric likelihood ratios. The possible combinations for computing similarity between test and reference data using the residual error are shown in Figure 4.3. As can be seen that by reversing the sense of test and reference in the log likelihood ratio we obtain another ratio, δ'/α', which is computed in exactly the same way as in Equations (4.27) and (4.29). with primes added as appropriate.

If a maximum likelihood formulation to linear prediction is used making the assumptions that the speech was generated by a Gaussian process passed through an all-pole filter and the analysis window length is much greater than the filter order the following integral results

$$\Xi = \int_{-\pi}^{\pi} \left[e^{V(\theta)} - V(\theta) - 1 \right] \frac{d\theta}{2\pi}. \tag{4.32}$$

Gray and Markel show that this integral can be written in terms of the likelihood ratios as

$$\Xi = (\sigma/\sigma')^2 (\delta/\alpha) - 2 \ln(\sigma/\sigma') - 1. \tag{4.33}$$

If the gains are taken to be equal we obtain

$$\delta/\alpha = 1 + \Xi. \tag{4.34}$$

If the roles of the test and reference are reversed, see Figure 4.2, we can write the reverse likelihood ratio, switching the primes

$$\Xi' = (\sigma'/\sigma)^2 (\delta'/\alpha') - 2\ln(\sigma'/\sigma) - 1, \tag{4.35}$$

and with equal gains,

$$\delta'/\alpha' = 1 + \Xi' \tag{4.36}$$

where

$$\Xi = \int_{-\pi}^{\pi} \left[e^{-V(\theta)} + V(\theta) - 1 \right] \frac{d\theta}{2\pi}. \tag{4.37}$$

Note that changing the roles of test and reference spectrum is equivalent to replacing $V(\theta)$ with $-V(\theta)$.

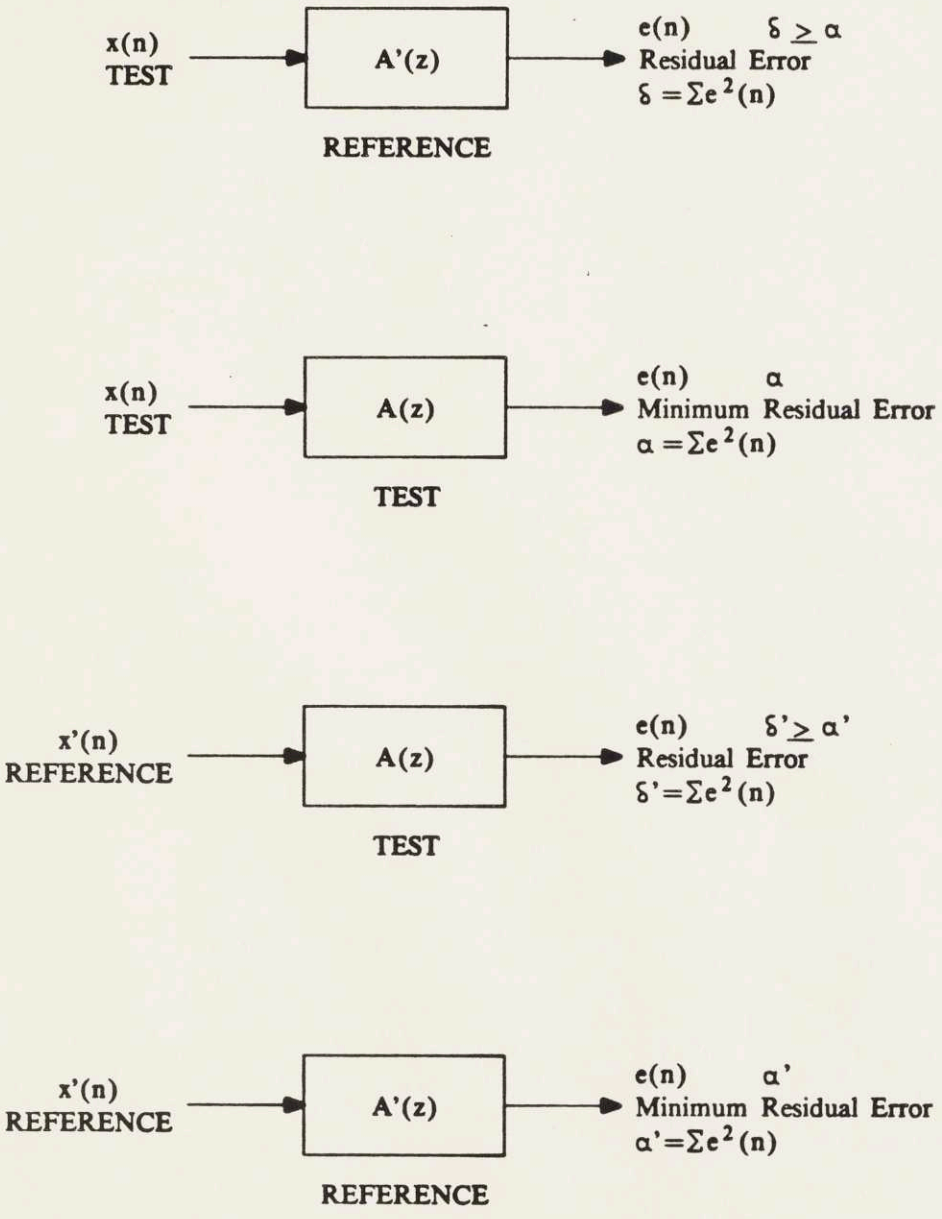


Figure 4.3. Possible Combinations of Test and Reference Data

A symmetric distance measure can be arrived at by simply averaging the two nonsymmetrical terms Ξ and Ξ'. This is written as

$$\Omega = \frac{1}{2}(\Xi + \Xi') = \int_{-\pi}^{\pi} \left\{ \cosh[V(\theta)] - 1 \right\} \frac{d\theta}{2\pi}. \tag{4.38}$$

This is arrived at by averaging the integrals of (4.32) and (4.37). Substituting the expressions for Ξ and Ξ' in Equations (4.33) and (4.35) respectively we obtain the relation

$$\Omega = \frac{1}{2}(\sigma/\sigma')^2(\delta/\alpha) + \frac{1}{2}(\sigma'/\sigma)^2(\delta'/\alpha') - 1. \tag{4.39}$$

If the gains are taken to be equal, Ω reduces to

$$\Omega = \frac{\delta/\alpha + \delta'/\alpha'}{2} - 1, \tag{4.40}$$

the arithmetic mean of the likelihood ratios minus one. The value of Ω is related to the decibel scale by defining ω in terms of the inverse function used in the integral

$$\cosh(\omega) - 1 = \Omega \tag{4.41}$$

solving for ω we have

$$\omega = \ln\left[1 + \Omega + \sqrt{\Omega(2+\Omega)}\right]. \tag{4.42}$$

The measure given by ω is referred to as the cosh distance measure due to the form of the integrand of Equation (4.38) and can be efficiently computed using the likelihood ratios and Equations (4.40) and (4.42). The cosh measure, ω, is converted to decibels by multipling by the factor 4.3429. It can be shown that the cosh distance measure is always larger than the rms measure d_2 and approaches it for small values of $V(\theta)$.

4.4.3 The Cepstral Distance Measure

Consider the expression for the L_2 norm

$$(d_2)^2 = \int_{-\pi}^{\pi} |V(\theta)|^2 \frac{d\theta}{2\pi}. \tag{4.43}$$

Using Parseval's theorem and the relationship given in Equation (4.9) this measure can be written

as

$$(d_2)^2 = \sum_{k=-\infty}^{\infty} (c_k - c'_k)^2. \tag{4.44}$$

Unfortunately this relationship only demonstrates that the L_2 norm can be computed from the cepstral coefficients if an infinite number of terms are included. Gray and Markel suggest the use of a truncated series to compute an approximation to $(d_2)^2$. They refer to this cepstral distance measure as $u(\mathrm{L})$, which is written as

$$[u(\mathrm{L})]^2 = \sum_{k=\mathrm{L}}^{\mathrm{L}} (c_k - c'_k)^2 \tag{4.45}$$

$$= (c_0 - c'_0)^2 + 2\sum_{k=1}^{\mathrm{L}} (c_k - c'_k)^2$$

where L is the number of terms in the series. The cepstral distance measure $u(\mathrm{L})$ can be interpreted as the rms distance between the spectral models after each has been homomorphically smoothed to L terms. As the number of terms increases, $u(\mathrm{L})$ approaches d_2 from below and in the limit

$$\lim_{\mathrm{L}\to\infty} u(\mathrm{L}) = d_2. \tag{4.46}$$

This measure provides a convenient method of approximating the rms log spectral measure, d_2.

The question of how many cepstral coefficients are necessary to adequately represent the spectral model was addressed by Gray and Markel[18]. Since the first p cepstral coefficients, not including the gain term c_0, uniquely determine the filter coefficients a_k, it is necessary that L be greater than or equal to p. If L is less than p the positive definiteness property of the distance measure is destroyed. Their work shows that even for the minimum number of coefficients, $\mathrm{L} = p$, the $u(\mathrm{L})$ distance measure is strongly correlated to the L_2 norm, with a correlation coefficient of 0.98.

4.5 Summary

In this chapter we have discussed the LPC and LPC derived cepstral speech parameters that were chosen for our system. The means of computing these parameters, and their averages, has

been covered as well as the distance measures over these parameter sets that were considered. In the following chapter we will discuss the experiments undertaken and the results of these experiments.

5. Experimental Work and Results

In the preceding chapters we have discussed the concept of segmentation for data reduction and presented a generalized acoustic segmentation system. The principles of Dynamic Time Warping and the mathematical formalism used by Myers et.al.[8] were covered in some detail. Three methods of incorporating durational information into the DTW algorithm were presented. We also discussed the methods of computation of the speech parameters and the distance measures used in this work. In this chapter we will discuss the experimental work that was carried out to demonstrate these concepts and present the results.

5.1 The Test Corpus

The vocabulary used in all our experiments was the 30 word calculator task shown in Table 5.1. This vocabulary is of a reasonable size and provides several easily confused words, such as "store" and "four", or "A" and "eight". This vocabulary was chosen as opposed to the English alpha-digit vocabulary, because it contains predominately polysyllabic words and better demonstrates the advantages that are gained from data reduction by acoustic segmentation. Since many of the words of the alpha-digit vocabulary are simple consonant-vowel combinations, the calculator task provides a wider range of phonetic transitional characteristics and is a more rigorous test of the segmentation system.

Four speakers were used in these experiments, two female (CAB, JAJ) and two male (CDB, RDB). Six repetitions of the calculator task were recorded for each speaker and were recorded in two sessions of three repetitions each. The recording sessions were separated by a period of a week in an attempt to avoid speaker bias that might arise from repetition within a short time interval. The vocabulary list was randomized prior to the recording sessions to prevent particular word sequences from influencing speaker pronunciation. The speech was recorded in a quiet room with a Sony lapel microphone. The speech was subsequently processed by the system shown in Figure 5.1 and described here. The analog speech signal $s(t)$ was lowpass filtered to 4.8 KHz, sampled at 10 KHz, and linearly quantized to 12 bits to obtain the digitized signal $s(n)$. The digitized

WORD	#PHONEMES	ARPABET TRANSLATION
A	1	EY
B	2	B IY
add	2	AE D
cosine	5	K OW S AY N
degrees	6	D IX G R IY Z
delete	5	D IH L IY T
divide	5	D IX V AY D
eight	2	EY T
equals	5	IY K W EL Z
exponent	9	EH K S P OW N AX N T
five	3	F AY V
four	3	F OW R
inverse	5	IH N V ER S
minus	5	M AY N AX S
multiply	8	M AH L T IX P L AY
nine	3	N AY N
one	3	W AH N
plus	4	P L AH S
radians	7	R EY D IY AX N Z
recall	5	R IY K AO L - R IX K AO L
reciprocal	9	R IX S IH P R A K EL
seven	5	S EH V EH N
sine	3	S AY N
six	4	S IH K S
store	4	S T AO R
subtract	8	S AH B T R AE K T
tangent	7	T A NX JH EH N T
three	3	TH R IY
two	2	T UW
zero	4	Z IY R OW

TABLE 5.1. Calculator Task Vocabulary

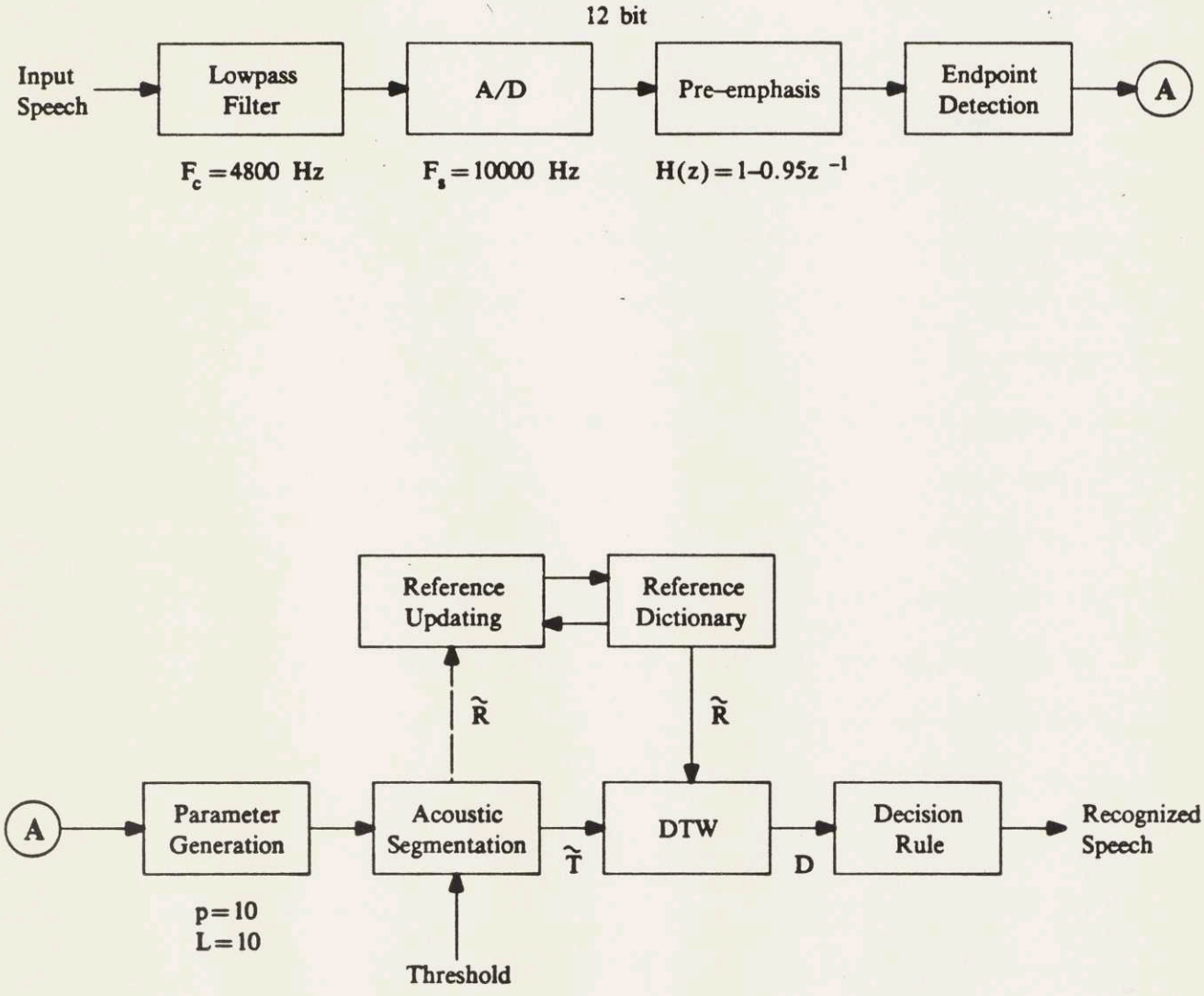


Figure 5.1. Segmentation and Isolated Word Recognition System

speech signal was pre-emphasized with a pre-emphasis factor of 0.95. The endpoints of the utterances were determined using the endpoint detection system described by Lamel et.al.[28]. No attempt was made to correct errors in the endpoints as it was felt that this procedure would more closely approximate a true operating environment for the speech recognition system. For some speakers the stop burst from word final stop consonants would occasionally be missed. This did not, however, present a significant problem to the word recognition system. After the endpoints were determined, a tenth order LPC parameter vector was generated every 15 ms using a 25.6 ms Hamming window. If a cepstral representation was desired these LPC coefficients were converted using the recursive relation described in Chapter four to obtained the first eleven cepstral coefficients, c_k, $k=0,1,\ldots 10$. The parametric representation was then segmented and compressed using the acoustic segmentation system discussed in Chapter two. The threshold for the segmentation was determined from statistics for the distance measure over the entire corpus. The reference dictionary was speaker dependent, with the first repetition for each speaker arbitrarily chosen as the reference set and the remaining five were used as test data. The Dynamic Time Warping algorithms used for these experiments are those specified in chapter three. The Type I local continuity constraints and the weighting function discussed in chapter three were used for all experiments. The decision rule was simply choosing the word associated with the minimum total distance. The entire system was implemented on a Digital Equipment Corporation PDP-11/60 minicomputer in the "C" programming language. Before we discuss the experimental work we should discuss the performance measures that were used to evaluate the DTW algorithms.

5.2 Measures of Relative Performance

There are several measures of performance which must be considered in evaluating an isolated word recognition system. First and foremost is the recognition accuracy. A system which cannot accurately recognize the input speech is of little use no matter how efficient in computation and storage it may be. The higher the recognition accuracy, or conversely the lower the error rate, the better the recognition system. We will use percent error to express this measure of performance. Percent error is written

$$percent\ error = \frac{\#\ of\ incorrect\ guesses}{total\ \#\ of\ guesses} \cdot 100\%. \tag{5.1}$$

Computation time is another performance measure that is of great importance. If the computation time is excessive then the vocabulary size may have to be restricted to allow recognition within a reasonable amount of time, limiting the usefulness of the recognition system. The most useful measure of computation time is the average time per word compare, or warp. This is usually expressed as seconds per warp. A third measure of performance is the the amount of storage required for the speech templates. This factor can also limit the vocabulary size in a speech recognition system by placing restrictions on the size of the reference dictionary. The storage requirement, when considering data reduction, can be expressed as a percentage of the amount of data used in the normal case. We will use percent compression as the measure of performance in minimizing storage. Percent compression is written as

$$percent\ compression = \frac{\#\ of\ frames - \#\ of\ segments}{\#\ of\ frames} \cdot 100\%. \tag{5.2}$$

In actuality the percent compression will be slightly less than this, since the segment duration component is added to the segmented representation. However, this information can be encoded in a relatively small number of bits and does not significantly effect the amount of storage required. It is interesting to evaluate recognition accuracy in terms of percent compression and computation time in order to determine what tradeoffs exist between these three factors. In the following section we discuss the experiments that were performed and the results of these.

5.3 Experiments and Results

Four different experiments were carried out for this thesis. The first of these was a comparison of the performance of the three distance measures. This experiment represents the best performance in terms of recognition accuracy we can expect for isolated word recognition using previously tested methods. The second experiment tests the performance of the Method 1 algorithm for handling durational information, which ignores the durational component of the segmented templates. The third experiment tests the Method 2 algorithm for handling durational information by weighting the local distance by the average of the segment lengths. The fourth experiment tests the

Method 3 for handling durational information, which expands the segments to their original length. If our methods of improving storage and computation give recognition score as good or better than for the first experiment then the use of these methods is clearly advantageous.

EXPERIMENT I: This experiment was conducted to determine the relative performance of the three distance measures presented by Gray and Markel in isolated word recognition for the unsegmented case. Each distance measure was used for the local distance $d(\mathbf{R}(n),\mathbf{T}(m))$ in the normal DTW algorithm. The normal DTW algorithm, as was covered in chapter three, is specified as follows: the local distance is given by

$$\tilde{d}(n,m) = d(\mathbf{R}(n),\mathbf{T}(m)) \tag{5.3a}$$

the normalization factor by

$$\mathbf{N}(\tilde{W}) = N+M \tag{5.3b}$$

and the minimum total distance by

$$D = \frac{D_A(N,M)}{\mathbf{N}(\tilde{W})}. \tag{5.3c}$$

As was previously stated no segmentation was used in this experiment, i.e. the threshold was set to zero. Note that as the threshold is lowered the amount of data compression that occurs is reduced and the representation naturally degenerates to the unsegmented case as the threshold approaches zero. This experiment sets our baseline for computation, storage, and recognition accuracy and the results from the remaining experiments can be evaluated in relation to this experiment. This experiment was conducted for each speaker using the first repetition of the vocabulary as the reference and the remaining five as the test data. The error rates, in percent error, for each distance measure are given in Table 5.2 and are also shown in Figure 5.2.

As can be seen from these results the cosh distance measure gave the lowest error rate overall The cepstral distance measure performed slightly poorer and the log likelihood ratio performed poorest of the three. These error rates correspond well with previous work done in isolated word recognition. Our criterion for deciding which distance measure was to be used in the remaining

speaker	distance measure		
	cosh	cepstral	Itakura
CAB(f)	3.33	3.33	6.67
CDB(m)	2.04	4.08	2.68
JAJ(f)	1.33	2.00	4.00
RDB(m)	0.00	0.00	0.68
overall	1.69	2.36	3.53

TABLE 5.2. Percent Error for Three Distance Measures, No Segmentation.

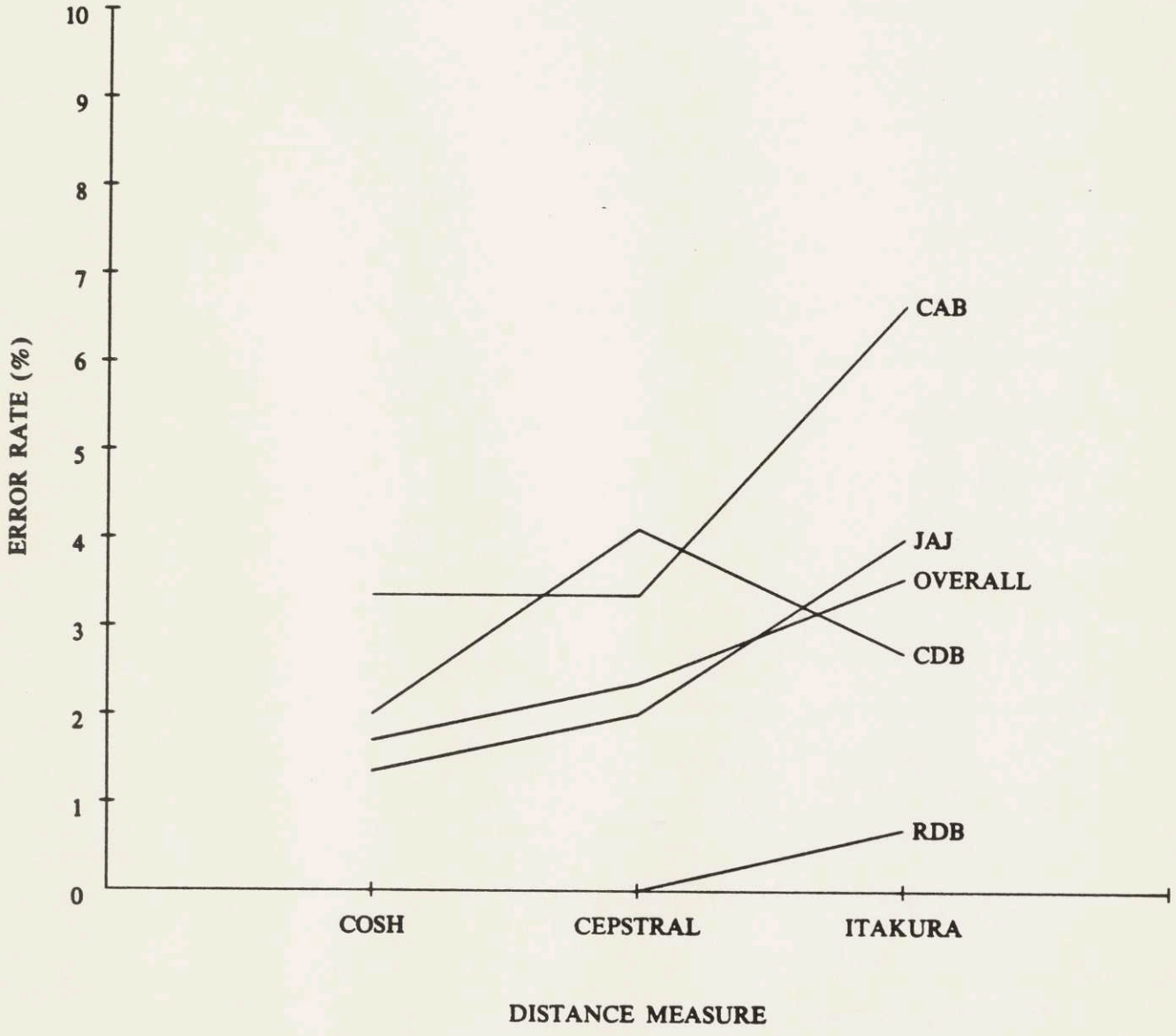


Figure 5.2. Percent Error vs. Distance Measure

experiments was the error rate alone. Consequently the cosh distance measure was chosen for experiments in segmentation, Although similar results for the remaining experiments could be expected for the other distance measures.

The remaining experiments compare the performance of three methods of handling durational information for three levels of data compression, 50%, 75%, and 90%. In order to determine the necessary thresholds for these compression ratios a histogram of the cosh distance contours for the entire corpus was made. This histogram is shown in Figure 5.3. The histogram was used to determine the threshold at which the ratio of number of times the distance, $d(\mathbf{R}(m),\mathbf{C}(m))$, exceeds the threshold to the total number of distance points yields the desired compression. Using this approach it was determined that a threshold of 5.75 dB gave approximately 90% compression, 3.75 dB gave approximately 75% compression, and 2.5 dB gave approximately 50% compression. Table 5.3 gives the actual compression ratios that were obtained for these thresholds. Note that the actual percent compression is less than predicted from the histogram. This was due to the fact that the number of segments in an utterance is always the number of times the distance contour exceeds the threshold plus one, there must always be at least one segment per utterance. This fact was not taken into account in determining the thresholds and the compression ratios that are shown in Table 5.3 are those that were used for all the remaining experiments.

EXPERIMENT II: In chapter three it was argued that recognition accuracy could be improved by ignoring the durational component in the segmented templates. This experiment tests this hypothesis by ignoring the durational component and applying the DTW algorithm directly to parameter vectors of the segmented templates. This method was implemented by using the specifications discussed in chapter three for Method 1, the local distances are given by

$$\tilde{d}(n,m) = d(\tilde{\mathbf{R}}(n),\tilde{\mathbf{T}}(m)), \tag{5.4a}$$

the normalization factor

$$\mathbf{N}(\tilde{W}) = S^{R}_{MAP}(N) + S^{T}_{MAP}(M), \tag{5.4b}$$

and the minimum total distance

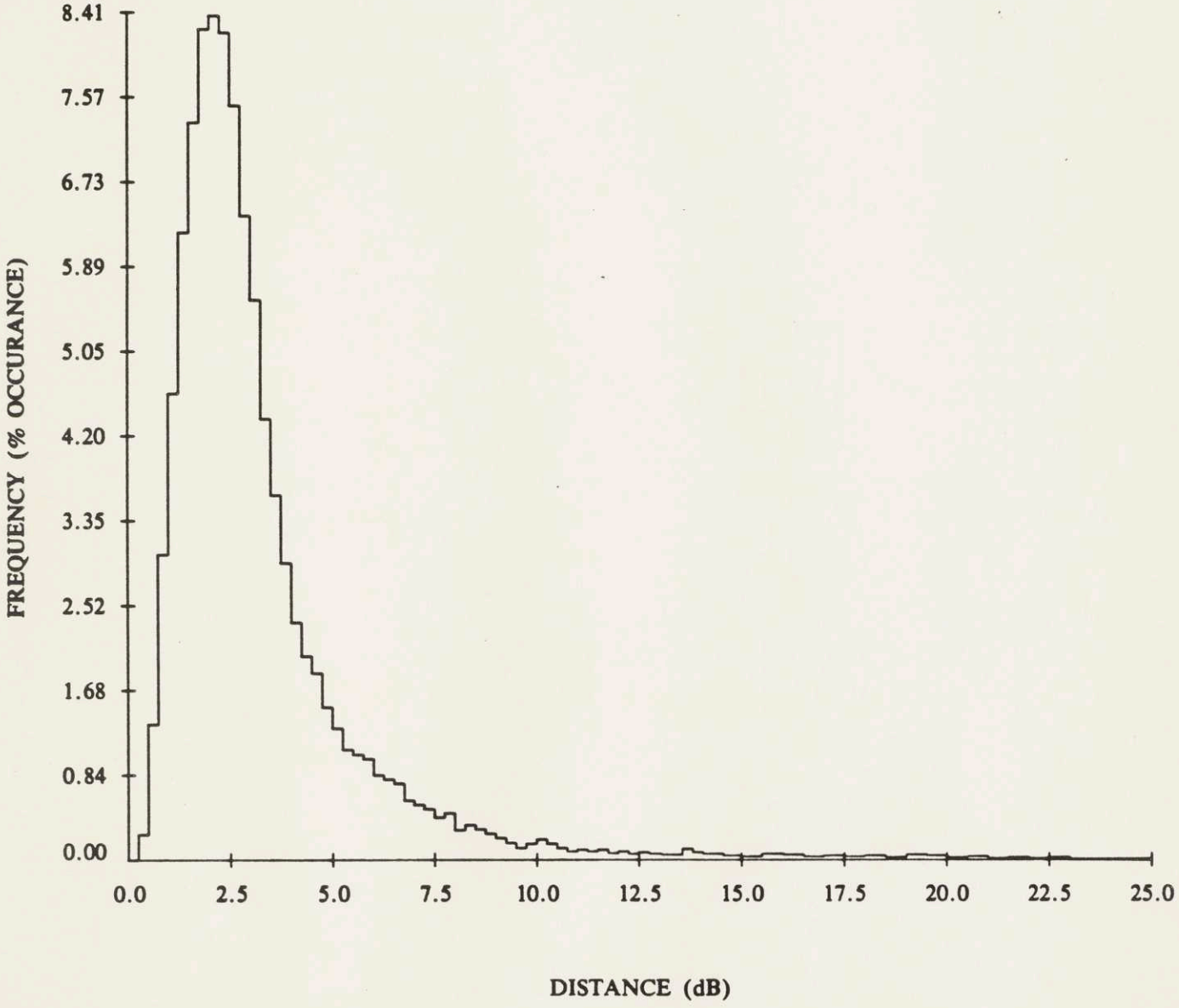


Figure 5.3. Histogram of Cosh Distance Measure

speaker	threshold (dB)		
	5.75	3.75	2.50
CAB(f)	88.3	74.8	49.9
CDB(m)	87.7	70.6	41.5
JAJ(f)	87.7	73.8	44.8
RDB(m)	86.9	75.9	52.1
overall	87.7	73.6	46.8

TABLE 5.3. Actual Percent Compression for Three Thresholds

$$D = \frac{D_A(S^R_{MAP}(N), S^T_{MAP}(M))}{\mathbf{N}(\tilde{W})}. \tag{5.4c}$$

This experiment was conducted for each speaker using the first repetition as the reference and the remaining five as test data. The error rates, in percent error, for each of the compression rates for Method 1 are given in Table 5.4.

As can be seen from these results error rate increases significantly with percent compression. Even for 50% compression the overall error rate has increased more than 5% over the unsegmented case. This is considerably worse than indicated by previous work[12]. One reason for the high error rate is that a considerable amount of information is contained in the segment durations and that ignoring this information reduces recognition accuracy. Since the parameter vectors in the segmented templates generally represent more than one frame of speech data, the contribution of the local distance between test and reference segments is a significantly smaller factor than is actually required for more accurate recognition. In the following experiment Method 2 is used to improve recognition accuracy by weighting local distances in the DTW algorithm on the basis of segment duration.

EXPERIMENT III: In this experiment Method 2 is used to incorporate durational information into the solution, i.e. the local distances computed between parameter vectors of the segmented test and reference templates are multiplied by the average of the durational components. The specification for this method, as discussed in chapter three, is given by the local distances

$$\tilde{d}(n,m) = \frac{L_R(n)+L_T(m)}{2} \cdot d(\tilde{\mathbf{R}}(n),\tilde{\mathbf{T}}(m)), \tag{5.5a}$$

the normalization factor

$$\mathbf{N}(\tilde{W}) = N + M, \tag{5.5b}$$

and the total accumulated distance is

$$D = \frac{D_A(S^R_{MAP}(N), S^T_{MAP}(M))}{\mathbf{N}(\tilde{W})}. \tag{5.5c}$$

This approach does not significantly increase the amount of computation over Method 1, yet it still

speaker	percent compression			
	0%	50%	75%	90%
CAB(f)	3.33	10.7	20.7	38.7
CDB(m)	2.04	4.00	11.3	35.3
JAJ(f)	1.33	4.67	21.3	45.3
RDB(m)	0.00	7.53	17.8	38.4
overall	1.69	6.71	17.8	39.4

TABLE 5.4. Percent Error for Segmented Templates, Method 1

incorporates the durational information into the solution. Table 5.5 shows the percent error for each of the compression rates using this method.

The results obtained from weighting the local distances did not improve the recognition accuracy and for the most part gave poorer recognition results than the previous method. A possible explanation of the poor performance of both these methods is that the nonlinear compression brought about by the segmentation algorithm causes the required time alignment path to be more drastic than for the unsegmented case. Consequently regions of similar acoustic characteristics in the two segmented templates may fall outside the allowable range or slope constraints for time alignment in the DTW algorithm. The weighting function used for Method 2 based on segment durations does not effect the local or global range constraints and thus cannot improve recognition accuracy.

EXPERIMENT IV: Method 3 eliminates the nonlinear time compression from the segmented templates by expanding the segments to their original length. This was achieved by repeating the segmental average the appropriate number of times, i.e.

$$\tilde{\mathbf{R}}'(n) = \tilde{\mathbf{R}}(S^R_{MAP}(n)), \qquad n=1,2,\ldots,N \tag{5.6a}$$

$$\tilde{\mathbf{T}}'(m) = \tilde{\mathbf{T}}(S^T_{MAP}(m)), \qquad m=1,2,\ldots,M \tag{5.6b}$$

Dynamic Time Warping was then carried out on the expanded templates. While this method requires more computation than the previous two, it still gives savings over the unsegmented case. The Method 3 DTW algorithm is specified by the local distance

$$\tilde{d}(n,m) = d(\tilde{\mathbf{R}}(S^R_{MAP}(n)), \tilde{\mathbf{T}}(S^T_{MAP}(m))), \tag{5.7a}$$

the normalization factor

$$\mathbf{N}(\tilde{W}) = N + M, \tag{5.7b}$$

and the minimum total distance is

$$D = \frac{D_A(N,M)}{\mathbf{N}(\tilde{W})}. \tag{5.7b}$$

speaker	percent compression			
	0%	50%	75%	90%
CAB(f)	3.33	15.3	25.3	40.0
CDB(m)	2.04	5.33	14.7	36.0
JAJ(f)	1.33	4.67	20.7	43.3
RDB(m)	0.00	7.53	13.7	30.8
overall	1.69	8.22	18.6	37.6

TABLE 5.5. Percent Error for Segmented Templates, Method 2

This is similar to the approach taken by Tappert and Das[12] in their work. This experiment was carried out for each speaker using the first repetition of the vocabulary as the reference and the remaining five as test data. Results from this experiment are shown in Table 5.6. This approach gives substantially better recognition accuracy than the previous two methods and the results agree more closely with those given by Tappert and Das. Figure 5.4 shows the overall recognition performance of the three methods in percent error as a function of compression. From this figure we can see the general trend reflected in the error rate as it increases with percent compression. Error rates for Methods 1 and 2 increase at roughly the same rate, while the error rate for Method 3 increases more slowly but begins to accelerate above 75% compression. This indicates that the segmented representation begins to deteriorate quickly above 75% compression.

We have considered the performance of all methods and distance measures in terms of recognition accuracy as function of distance measure and percent compression. The performance of these algorithms should also be evaluated in terms of computation. Table 5.7 contains the average computation time in seconds per warp for each of the three methods, these results are also plotted in Figure 5.5. The amount of computation required for the first two methods drops off more quickly with compression than that required for the expansion method. However, the recognition accuracy is significantly better for this method.

Let us now consider the tradeoffs between computation and percent error. Figure 5.6 shows overall percent error for each of the three methods as a function of computation time. This demonstrates the trade off between computation and recognition accuracy, we can see that the break even point in terms of recognition accuracy for the three methods is somewhere about 7% error and 5 sec/warp. If one is willing to accept a slightly higher computation time Method 3 is definitely superior to either Methods 1 or 2.

5.4 Summary

In this chapter we have presented the experimental work carried out in support of this thesis. The results of this work demonstrate several points. The cosh distance measure was shown to give

speaker	percent compression			
	0%	50%	75%	90%
CAB(f)	3.33	3.33	6.00	19.3
CDB(m)	2.04	2.67	4.00	20.0
JAJ(f)	1.33	1.33	4.67	21.3
RDB(m)	0.00	1.37	4.11	21.9
overall	1.69	2.18	4.70	20.6

TABLE 5.6. Percent Error for Segmented Templates, Method 3

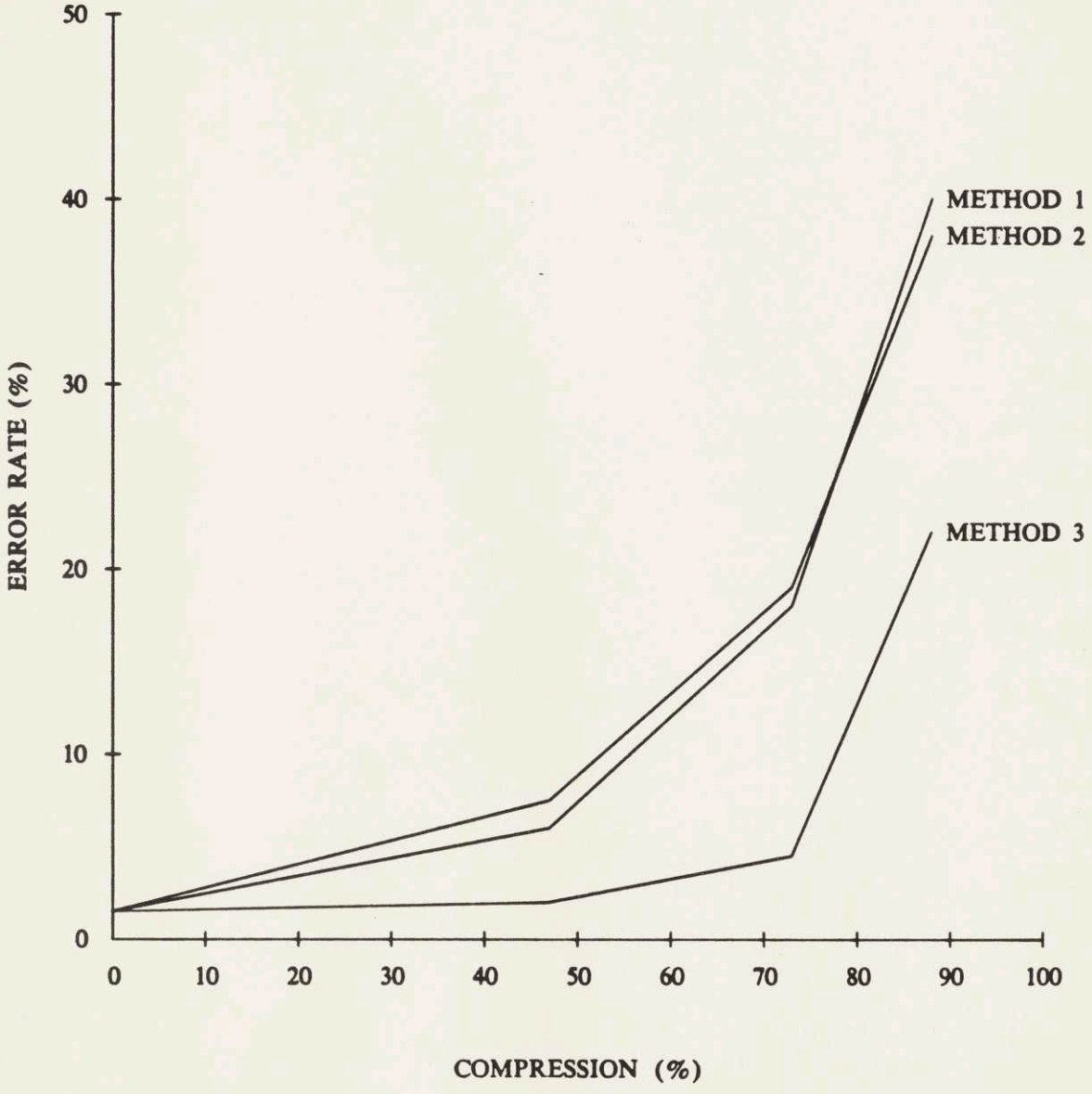


Figure 5.4. Percent Error vs. Percent Compression

method	percent compression			
	0%	50%	75%	90%
1	17.3	4.17	0.89	0.18
2	17.3	4.26	0.89	0.18
3	17.3	7.76	5.04	4.37

TABLE 5.7. Average Computation Time (sec/warp) for Three Methods of Handling Duration

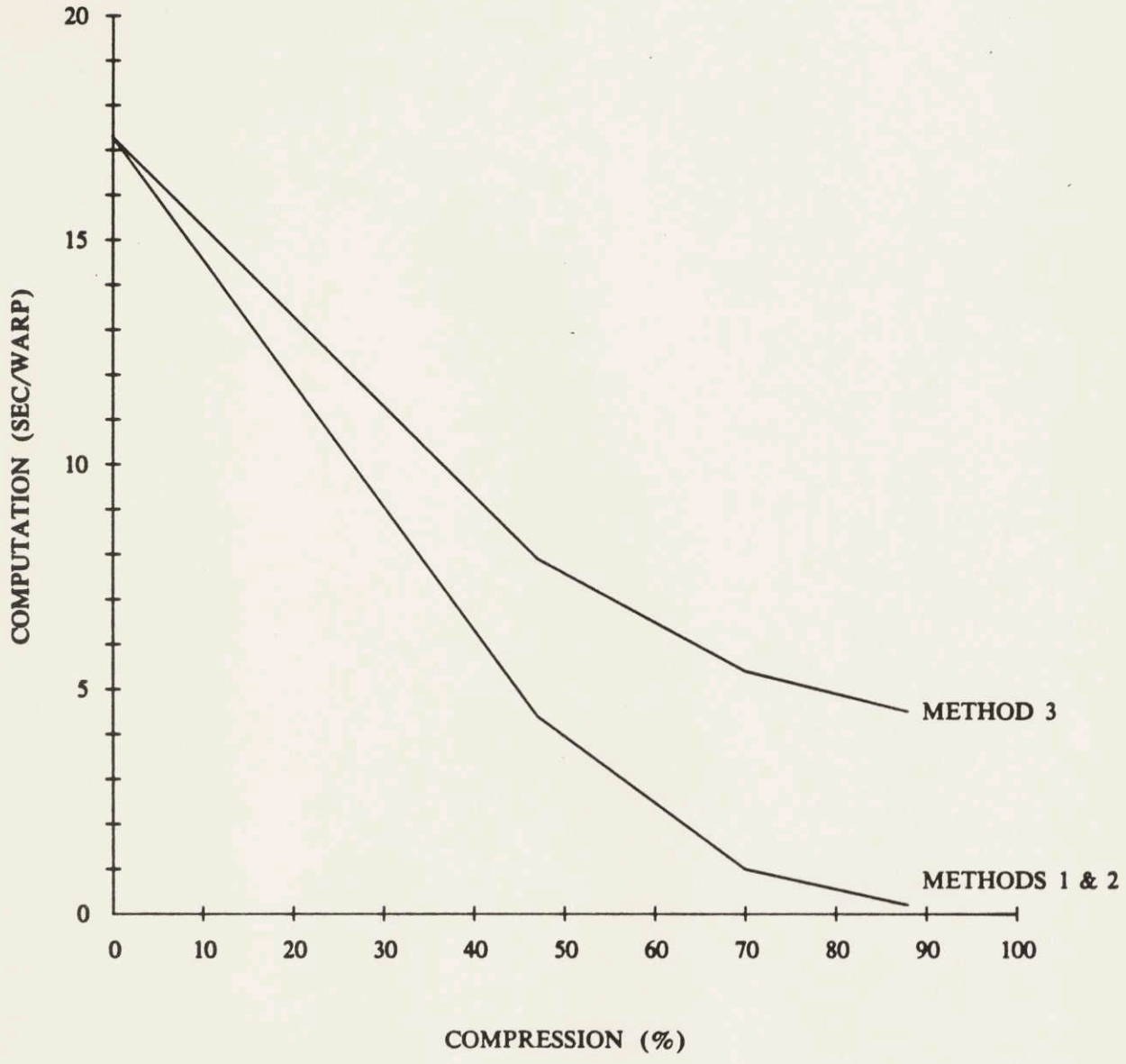


Figure 5.5. Computation Time vs. Percent Compression

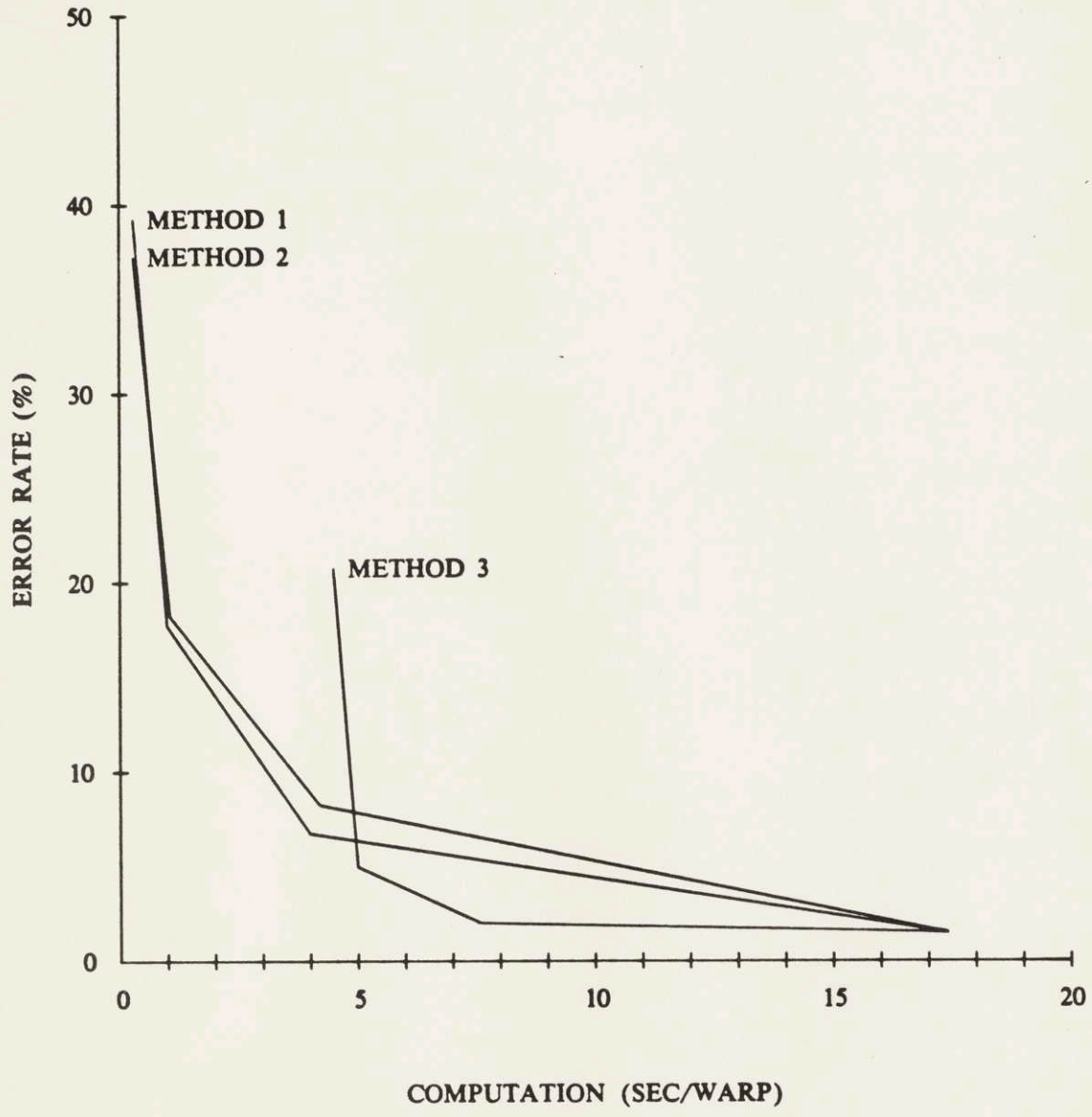


Figure 5.6. Percent Error vs. Computation Time

better recognition performance than either the log likelihood ratio or the cepstral distance measure in isolated word recognition using the normal DTW algorithm. None of the methods of handling segment durations tested gave better recognition accuracy than the unsegmented case. Acoustic segmentation cannot give improvements in recognition accuracy, however, it does give improvements in computation and storage without significantly degrading recognition accuracy. Acoustic segmentation gave savings in computation and storage with out significantly degrading recognition accuracy only if the segments were expanded prior to recognition. This method gave a savings of 47% in storage and 52% in computation without significantly degrading recognition accuracy. This result is slightly poorer than those reported by Tappert and Das[12]. One reason for this is that the English alpha-digit vocabulary was used in their experiments and the arguments for improved recognition discussed in Chapter three are more applicable with the alpha-digit vocabulary. One possible explanation of this result is that the time alignment for segmented templates without expansion exceeds the slope and range constraints of the normal DTW algorithm. It is possible that relaxing the global range and slope constraints could improve recognition accuracy. Unfortunately it is likely that this would allow totally unreasonable time alignments, counteracting any improvement in recognition accuracy obtained from segmentation. Whatever means are used it is clear that segmental duration is an important factor in pattern matching techniques.

6. DISCUSSION

In the preceding chapters we have discussed acoustic segmentation for data reduction and demonstrated the performance improvements that can be obtained for isolated word recognition. In this chapter we will discuss how this technique could be useful for other speech recognition problems as well as make recommendations for future work. One point which should be emphasized is that the methods discussed in this thesis do not make any fundamental changes to the basic DTW algorithm and that it can easily be used in any system which makes use of this basic algorithm. Many applications of Dynamic Time Warping have been suggested aside from isolated word recognition. For example, techniques have been proposed for applying Dynamic Time Warping to word spotting[34] [35] and connected word recognition[36] [37]. The data compression techniques discussed in this thesis can be applied to these algorithms as well. Techniques for improving computation time of DTW algorithms, such as range limiting[9] or using the locally optimum path[11] do not change the basic DTW algorithm and can be applied to any of the methods discussed in chapter three to obtain further improvements in computation.

One implication of our work is that the use of this technique of data reduction could be used for savings in storage alone. In systems for which computation is not an issue, for example the multiprocessing array proposed by Ackland, et.al.[38] [39], the reference dictionary could be stored in compressed form and be expanded for comparison to the uncompressed input. Clearly if this is the case then any technique for bandwidth compression, such as variable frame rate vocoding[40], could equally be used. Experimental work should be undertaken to determine what savings in storage can be obtained using this type of approach.

6.1 Recommendations for Further Work

There are two areas in which future work could be suggested. The results of our experimental work indicate that this approach to data reduction will allow significant reductions in both storage and computation without sacrificing recognition accuracy. It is quite reasonable to expect that even greater savings can be effected if improvements were made in the segmentation and DTW

algorithms.

6.1.1 Suggestions for Segmentation

One factor that was not evaluated in our experimental work was the effect of the comparison frame of the segmentation algorithm. It would be useful to evaluate the impact of this choice on recognition accuracy in the DTW algorithm. One problem with using the adjacent frame as the comparison frame is that slow transitions will be missed and a segment with a rather large net acoustic change will be averaged giving a poor representation of the segment. The use of the average frame of the current segment so far while having greater overhead would most likely result in a better segmentation and consequently better recognition accuracy.

Another factor which plays an important role in the segmentation algorithm is the boundary decision. In our system the thresholding algorithm was chosen for simplicity and ease of implementation. The use of a more sophisticated boundary decision algorithm, such as a peak picking algorithm, could result in a better segmentation.

6.1.2 Suggestions for DTW Algorithms

As was discussed in chapter five one reason for the poor performance of Methods 1 and 2 was the effect of the nonlinear compression causing significant features of the two utterances to fall outside the global range constraints. In order to verify that this is indeed the case it would be worth while to test these methods using a local continuity constraint that had a greater range of slope. If this improved the recognition accuracy we could conclude that this was the reason for the poor recognition accuracy of Methods 1 and 2 in our experiments. One disadvantage of relaxing the slope constraints is that the number of paths to be traced is increased and the potential for unreasonable time alignments could arise.

One of the disadvantages of the Method 3 algorithm is its computational performance. Because the number of paths which must be traced is same as for the normal DTW algorithm the computational savings is not as great as for Methods 1 or 2. A possibility for improving the performance of this algorithm is to take advantage of the constant distance areas that occur in Method 3 from the

intersection of expanded segments. By proposing a modified local continuity constraint the number of paths which must be traced can be reduced.

Consider the constant distance area represented by the intersection of a test and reference segment shown in Figure 6.1. In this figure n_i and m_i are the initial frames for the reference and test segments respectively. The final frames for the reference and the test segments are n_f and m_f. In the recursive definition of the DTW algorithm a path could potentially enter this area anywhere along the top or right sides of this rectangular area and exit at any point along the bottom or left sides which falls within the angle formed by slope constraints, as shown in Figure 6.1. Using local continuity constraints as specified in chapter three there are many paths which must be traced through this region. Since the distance between test and reference is constant through out this region, the path through this rectangular area will be of constant slope. In an area of constant distance a linear path is optimum. It is not known apriori what the slope of the optimum path will be through this region be, so it is necessary to test all possible constant slope paths. The number of constant slope paths will be considerably less than the paths given by the local continuity constraints of chapter three. For larger segments the difference in the number of paths will be greater. If we specify a modified local continuity constraint which only tests the constant slope paths the amount of computation required for comparing two utterances is reduced.

If we apply a slope constraint to the time alignment path of

$$1 / E_{MAX} \leqslant slope \leqslant E_{MAX} \tag{6.1}$$

then the set of valid exit points (k, l) which can be reached from an entry point (n, m) is given by

$$\left\{ (k, l) \mid \left(m_i \leqslant l \leqslant \frac{(n - n_i)}{E_{MAX}} + m \text{ and } k = n_i \right) \text{ or } \left(n_i \leqslant k \leqslant E_{MAX}(n - n_i) + m \text{ and } l = m_i \right) \right\} \tag{6.2}$$

These merely represent the set of points along the lower and left sides of the constant distance area which fall within the angle formed by the slope constraints, as shown in Figure 6.1. We can specify this algorithm similarly to the way in which the other methods were specified in chapter three. In addition we must also specify the local continuity constraint. The local distance, for

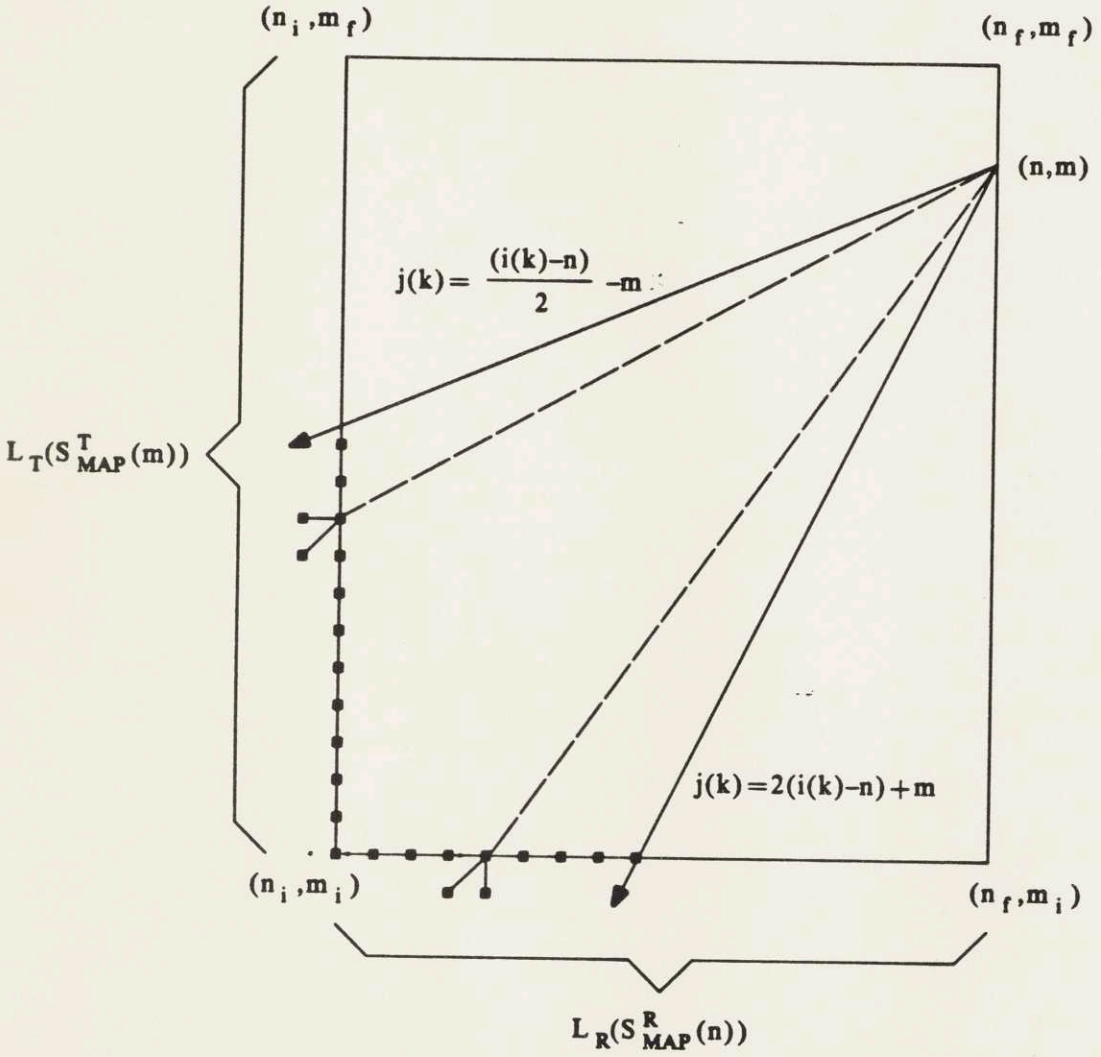


Figure 6.1. Constant Distance Area and Legal Range of Exit Points

efficiency, is given by the distance between the segmental representative vectors,

$$\tilde{d}(n,m) = d(\tilde{\mathbf{R}}(S^R_{MAP}(n)),\tilde{\mathbf{T}}(S^T_{MAP}(m))), \tag{6.3}$$

the recursive definition for the accumulated distance, or the local continuity constraints, can then be given by

$$D_A(n,m) =$$

$$\min \begin{bmatrix} D_A(k-1,l) + W(n,k,m,l)\cdot\tilde{d}(n,m) & \text{for } k=n_i,\ m_i \leqslant l \leqslant \dfrac{(n-n_i)}{E_{MAX}} + m \\ D_A(k-1,l-1) + W(n,k,m,l)\cdot\tilde{d}(n,m) & \text{for } k=n_i,\ m_i \leqslant l \leqslant \dfrac{(n-n_i)}{E_{MAX}} + m \\ D_A(k-1,l-1) + W(n,k,m,l)\cdot\tilde{d}(n,m) & \text{for } l=m_i,\ n_i \leqslant k \leqslant E_{MAX}(n-n_i)+m \\ D_A(k,l-1) + W(n,k,m,l)\cdot\tilde{d}(n,m) & \text{for } l=m_i,\ n_i \leqslant k \leqslant E_{MAX}(n-n_i)+m \end{bmatrix} \tag{6.4}$$

where $W(n,k,m,l)$ is a weighting function that is based on the length of the path through the constant distance area. In order to account for segments of one frame in length the weighting function should be

$$W(n,k,m,l) = \begin{cases} (n-k)+(m-l) & \text{if } L_R(n)\neq 1 \text{ and } L_T(m)\neq 1 \\ 1 & \text{otherwise} \end{cases} \tag{6.5}$$

This corresponds to the type d weighting function of Myers et.al.[8] which gives a normalization factor of

$$\mathbf{N}(\tilde{W}) = N + M. \tag{6.6}$$

Consequently the minimum total distance is given by

$$D = \frac{D_A(N,M)}{\mathbf{N}(\tilde{W})}. \tag{6.7}$$

While the local continuity constraint is significantly more complex the potential for computational savings remains. In order to implement this approach efficiently an additional component, the segment initial frame number n_i, must be added to the parameter vector for the segmented representation.

REFERENCES

1. A. Newell, J. Barnett, J. Forgie, C. Green, D. Klatt, J. C. R. Licklider, J. Munson, R. Reddy, W. Woods, "Speech-Understanding Systems: Final Report of a Study Group," Carnegie-Mellon Univ., Pittsburgh, PA, May 1971.

2. T. B. Martin, "Practical Applications of Voice Input to Machines," *Proceedings of IEEE,* Vol 64, No. 4, pp. 487-501, April 1976.

3. J. M. Nye, "The Expanding Market for Commercial Speech Recognizers," in W. A. Lea, ed., *Trends in Speech Recognition,* Prentice-Hall, Englewood Cliffs, N.J., pp. 461-468, 1980.

4. D. R. Reddy, "Speech Recognition by Machine: A Review," *Proceedings of IEEE,* Vol. 64, No. 4, pp. 501-531, April 1976.

5. L. R. Rabiner, "On Creating Reference Templates for Speaker Independent Recognition of Isolated Words," *Trans. Acoust., Speech, and Signal Proc.,* Vol. ASSP-26, No. 1, pp. 34-42, Feb. 1978.

6. H. Sakoe, S. Chiba, "A Dynamic Programming Approach to Continuous Speech Recognition," *Proceedings of International Conference on Acoustics,* Budapest, Hungary, Paper 20C-13, 1971.

7. F. Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition," *Trans. Acoust., Speech, and Signal Proc.,* Vol. ASSP-23, No. 1, pp. 67-72, Feb. 1975.

8. C. S. Myers, L. R. Rabiner, A. E. Rosenberg, "Performance Tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition," *Trans. Acoust., Speech, and Signal Proc.,* Vol. ASSP-28, No. 6, pp. 623-635, Dec. 1980.

9. H. Sakoe, S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *Trans. Acoust., Speech, and Signal Proc.,* Vol. ASSP-26, No. 1, pp. 194-200, Feb. 1978.

10. G. M. White, R. B. Neely, "Speech Recognition Experiments With Linear Prediction, Bandpass Filtering, and Dynamic Programming," *Trans. Acoust., Speech, and Signal Proc.,* Vol. ASSP-24, No. 2, pp. 183-188, April 1976.

11. L. R. Rabiner, A. E. Rosenberg, S. E. Levinson, "Considerations in Dynamic Time Warping Algorithms for Discrete Word Recognition," *Trans. Acoust., Speech, and Signal Proc.,* Vol. ASSP-26, No. 6, pp. 575-582, Dec. 1978.

12. C. C. Tappert, S. K. Das, "Memory and Time Improvements in a Dynamic Programming Algorithm for Matching Speech Patterns," *IEEE Trans. Acoust., Speech, and Signal Proc.,* Vol. ASSP-26, No. 6, pp. 583-586, Dec. 1978.

13. D. R. Reddy, P. J. Vicens, "A Procedure for the Segmentation of Connected Speech," *Jour. Audio Eng. Soc.,* Vol. 16, No. 4, pp. 404-411, Oct. 1968.

14. C. J. Weinstein, S. S. McCandless, L. F. Mondshein, V. W. Zue, "A System for Acoustic-Phonetic Analysis of Continuous Speech," *Trans. Acoust., Speech, and Signal Proc.,* Vol. ASSP-23, No. 1, pp. 54-67, Feb. 1975.

15. P. Mermelstein, "A Phonetic-Context Controlled Strategy for Segmentation and Phonetic Labeling of Speech," *Trans. Acoust., Speech, and Signal Proc.,* Vol. ASSP-23, No. 1, pp. 79-82, Feb. 1975.

16. R. Schwartz, J. Makhoul, "Where the Phonemes are: Dealing With Ambiguity in Acoustic-Phonetic Recognition," *Trans. Acoust., Speech, and Signal Proc.,* Vol. ASSP-23, No. 1, pp. 50-53, Feb. 1975.

17. D. H. Klatt, "SCRIBER and LAFS: Two New Approaches to Speech Analysis," in W. A. Lea, ed., *Trends in Speech Recognition*, Prentice-Hall, Englewood Cliffs, N.J., pp. 529-555, 1980.

18. A. H. Gray, Jr., J. D. Markel, "Distance Measures for Speech Processing," *Trans. Acoust., Speech, and Signal Proc.*, Vol. ASSP-24, No. 5, pp. 380-391, Oct. 1976.

19. H. G. Goldberg, D. R. Reddy, R. Suslick, "Parameter Independent Machine Segmentation and Labeling," *Conf. Proc. IEEE Symp. Speech Recognition*, Philadelphia, P.A., pp. 106-111, April 1974.

20. V. W. Zue, R. M. Schwartz, "Acoustic Processing and Phonetic Analysis," in W. A. Lea, ed., *Trends in Speech Recognition*, Prentice-Hall, Englewood Cliffs, N.J., pp. 101-124, 1980.

21. L. R. Rabiner, R. W. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall, Englewood Cliffs, N.J., 1978.

22. L. R. Rabiner, J. G. Wilpon, J. G. Ackenhusen, "On the Effects of Varying Analysis Parameters on an LPC-Based Isolated Word Recognizer," *Bell Syst. Tech. J.*, Vol. 60, No. 6, pp. 893-911, July 1981.

23. H. F. Silverman, N. R. Dixon, "A Comparison of Several Speech-Spectra Classification Methods," *Trans. Acoust., Speech, and Signal Proc.*, Vol. ASSP-24, No. 4, pp. 289-295, Aug. 1976.

24. R. Viswanathan, J. Makhoul, W. Russell, "Towards Perceptually Consistent Measures of Spectral Distance," *Conf. Rec. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, Philadelphia, P.A., pp. 485-488, April 1976.

25. D. H. Klatt, "A Digital Filter Bank for Spectral Matching," *Conf. Rec. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, Philadelphia, P.A., pp. 573-576, April 1976.

26. P. Mermelstein, "Automatic Segmentation of Speech into Syllabic Units," *J. Acoust. Soc. Am.*, Vol. 58, No. 4, Oct. 1975.

27. C. S. Myers *A Comparative Study of Several Dynamic Time Warping Algorithms for Speech Recognition*, Master's Thesis, Massachusetts Institute Of Technology, Feb. 1980.

28. L. F. Lamel, L. R. Rabiner, A. E. Rosenberg, J. G. Wilpon, "An Improved Endpoint Detector for Isolated Word Recognition," *Trans. Acoust., Speech, and Signal Proc.*, Vol. ASSP-29, No. 4, pp. 777-785, Aug. 1981.

29. L. R. Rabiner, M. R. Sambur, "Application of an LPC Distance Measure to the Voiced-Unvoiced-Silence Detection Problem," *Trans. Acoust., Speech, and Signal Proc.*, Vol. ASSP-25, No. 4, pp. 338-343, Aug. 1977.

30. L. R. Rabiner, M. R. Sambur, "An Algorithm for Determining the Endpoints of Isolated Utterances," *Bell Syst. Tech. J.*, Vol. 54, pp. 297-315, Feb. 1975.

31. J. D. Markel, A. H. Gray, Jr., *Linear Prediction of Speech* Springer-Verlag, New York, 1976.

32. A. V. Oppenheim, R. W. Schafer, *Digital Signal Processing* Prentice-Hall, Englewood Cliffs, N.J., 1975.

33. B. S. Atal, "Effectiveness of Linear Prediction Characteristics of the Speech Wave for Automatic Speaker Identification and Verification," *J. Acous. Soc. Am.*, Vol. 55, No. 6, pp. 1304-1312, June 1974.

34. R. W. Christiansen, C. K. Rushforth, "Detecting and Locating Key Words in Continuous Speech Using Linear Predictive Coding," *Trans. Acoust., Speech, and Signal Proc.*, Vol.

ASSP-25, No. 5, pp. 361-367, Oct. 1977.

35. C. S. Myers, L. R. Rabiner, A. E. Rosenberg, "On the Use of Dynamic Time Warping for Word Spotting and Connected Word Recognition," *Bell Syst. Tech. J.*, Vol. 60, No. 3, pp. 303-324, March 1981.

36. H. Sakoe, "Two-Level DP Matching - A Dynamic Programming Based Pattern Matching Algorithm for Connected Word Recognition," *Trans. Acoust., Speech, and Signal Proc.*, Vol. ASSP-27, No. 6, pp. 588-595, Dec. 1979.

37. C. S. Myers, L. R. Rabiner, "Connected Word Recognition Using a Level Building Dynamic Time Warping Algorithm," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Atlanta, G.A., pp. 951-955, April 1981.

38. B. Ackland, N. Weste, D. J. Burr, "An Integrated Multiprocessing Array for Time Warp Pattern Matching," *Proc. Int. Symp. Computer Architecture*, Minneapolis, MN, pp. 197-215, May 1981.

39. D. J. Burr, B. Ackland, N. Weste, "A High Speed Array Computer for Dynamic Time Warping," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Atlanta, G.A., pp. 951-955, April 1981.

40. V. R. Viswanathan, J. Makhoul, R. M. Schwartz, A. W. F. Huggins, "Variable Frame Rate Transmission: A Review of Methodology and Application to Narrowband LPC Coding," To Appear in *IEEE Trans. Communications*, April 1982.